Journal on Information Technologies & Communications
https://www.ictmag.vn
Volume 2025, Number 2
September

TABLE OF CONTENTS

Evidential Reasoning combine with Mass-based Similarity for Imbalanced Classification ... ***Anh Hoang*** 68

Innovation of Mechanical Practice Model using Virtual Reality Technology and Pilot Butt Welding technique in Flat Position ... ***Nguyen Van Huan*** 83

Enhancing Recommender Systems: A New Approach Using Collaborative Filtering with Bayesian Optimization and Gaussian Processes ... ***Tuan-Anh Nguyen*** 92

Efficient Mining of Concise Patterns for Frequent High-Utility Occupancy Itemsets Using Extended Pruning Strategies ... ***Tien Hoang, Lan Huynh, Hai Duong, Tin Truong*** 108

Distillation-Centric Approaches in Visual Question Answering with Mixture of Experts ... ***Huy Hoang Huynh, Tung Le, Huy Tien Nguyen*** 129

Biomedical Entity-Aware Semantic Role Labelling via Model Merging ... ***Hoai-Duc Tuan-Nguyen*** 140

An Approach for Dealing with Sequential Data in Intelligent Tutoring Systems ... ***Nguyen Xuan Ha Giang, Lam Thanh-Toan, Nguyen Thai-Nghe*** 151

Proposal of Dynamic Algorithms Combining a Substitution Layer and a Key Addition Layer for Block Ciphers using SHA3-256 ... ***Tran Thi Luong, Truong Minh Phuong*** 164

Evidential Reasoning combine with Mass-based Similarity for Imbalanced Classification

Anh Hoang
School of Business Information Technology, College of Technology and Design, University of Economics Ho Chi Minh City, Vietnam
Correspondence: Anh Hoang, Anhh@ueh.edu.vn
Communication: received 22/02/2025, revised 26/03/2025, accepted 08/04/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1358

***Abstract*: This paper introduces an integrated approach to address imbalanced classification problems using the Dempster-Shafer theory of evidence and mass-based dissimilarity measurement. Traditional distance-based and density-based classifiers often struggle with skewed datasets, leading to two key challenges: (1) misclassification bias, where conventional classifiers treat all instances equally despite the dominance of majority-class samples, and (2) variability in instance densities, which affects similarity assessments. To overcome these limitations, we introduce EMass, a novel classifier that replaces distance- and density-based similarity calculations with mass-based measures. Each neighbor of a considering instance is treated as an independent source of prior knowledge, represented through a basic probability assignment (BPA). We then apply Dempster's rule of combination to aggregate multiple sources of information, producing a comprehensive probability estimate for classification. Experimental evaluations are conducted on 60 imbalanced data sets, comparing 12 algorithms using key performance metrics, including the F1 score, the Brier score, and the area under the curve (AUC). The results show the effectiveness of the proposed EMass classifier in improving the classification performance for imbalanced data sets.**

Keywords: *Imbalance classification, Mass-based similarity, Multi-source evidence, Predictive modeling, Knowledge discovery.*

I. Introduction

Predicting classification model involves determining to which predefined categorical class a specific instance belongs. To date, numerous classification approaches have been proposed and applied successfully to a wide range of domains, such as Decision tree (DT) classifiers or DT classifier that integrates building and pruning [1], DT with AdaBoost [2]; Classification based on association rules (CBA) or multiple association rules (CMAR) [3], predictive association rules (CPAR) [4]; Bayesian classifiers applied Bayes' Theorem [5]; Lazy learning algorithms for classification, k-nearest neighbor algorithm (k-NN) [6]; Discriminative pattern mining for effective classification [7]; the Ensemble methods, such as random forest classifiers [8], rainforest framework [9], or XGBoost (XGB) [10]; Support vector machine (SVM) classifiers, Gaussian RBF kernel SVM (RBF_SVM) [11], or using SVM with hierarchical clustering for classifying large dataset [12]; and applying Neural network with feed-forward or back-propagation classifier [13]; Naïve associative classifier (NAC) [14], Assisted classification of imbalanced datasets (ACID) [15], and Rule-based evolutionary online learning systems (LCSs) [16] were specially designed for mixed and incomplete classification problems.

However, most of these methods are designed to focus merely on balanced datasets; therefore, they may not work effectively on imbalanced datasets where the class distributions are skewed. The challenge of classification with imbalanced datasets remains a critical and complex issue in machine learning, data mining, and knowledge discovery, as it significantly impacts model performance and decision-making accuracy. When datasets have a disproportionate ratio of objects in their classes, some classes, called minority classes, have only very small numbers of data instances, while the others, called majority classes, contain large numbers of data instances. Classification with such imbalanced data sets has been a challenging task for most of the conventional learning methods mentioned above, which assume a relatively balanced class distribution and uniform misclassification costs [17]. Furthermore, a disproportionate distribution of data between classes in imbalanced datasets also poses another important issue when distance- or density-based classification methods are applied due to context-dependent variation, as briefly discussed in Section II.2.

The above challenges need either to restructure the data set or to specialize the algorithm to address the imbalanced classification problem. Previous studies often focus on four groups of methods to fulfill these requirements: resampling the data space, algorithmic modifications, cost-

sensitive learning, and ensemble methods. First, resampling techniques consist of oversampling, such as the synthetic minority oversampling technique (SMOTE) and its variations based on how synthetic samples are generated; and the undersampling technique, such as the TOMEK links algorithm Both techniques transform and restructure the data space to decrease the impacts of their class distribution. Second, algorithms-oriented approaches customize existing algorithms [18] or develop new algorithms for the class imbalance task. Third, cost-sensitive learning solutions, such as the BEE-Miner algorithm [19] and the integration of TANBN with the cost-sensitive classification algorithm [20], aimed at incorporating the algorithm-oriented into the data-level methods to minimize the total misclassification costs, instead of trying to optimize the accuracy. Finally, yet importantly, ensemble learning models, for example multi-imbalance software [21], are operated by adapting the previous ensemble learning algorithms or embedding a cost-sensitive algorithm in the ensemble learning procedure. Notice that all of these learning approaches consume more resources to learn the parameters for the majority class than the minority class. The reason is because the treatments for all instances are the same.

The mass-based similarity measurement and the Dempster-Shafer theory of evidence strongly motivated us to study the problem of class imbalance in advanced. For example, in [22], the authors developed a multiclassifier approach that integrated the support vector machine and the Dempster-Shafer theory to support risk analysis. In [23], the authors introduced an algorithm for making decisions based on kernel density estimation and using a pairwise learning method for binary classification problems. In [24], the Dempster-Shafer theory of evidence (DST) based classifier was proposed with a new rule of evidential reasoning for combining multiple sources of information to construct a multiattribute classifier. To our knowledge, our study has been tested for the first time. The Dempster-Shafer framework is used for the imbalanced classification problem. Through this work, we propose a novel classifier approach for classifying imbalanced datasets, herein referred to as EMass. Each neighbor of the considering instance is treated as an independent evidence source, contributing evidence toward determining the target variable. The Dempster rule of combination is then applied to aggregate these multiple sources of evidence, enabling a more robust decision-making process. By incorporating mass-based measures, the proposed EMass approach handle the disadvantages of traditional distance- and density-based classifiers. Unlike conventional methods, this new classifier relaxes the constraints imposed by distance axioms and assumptions of data independence, improving its adaptability to complex and imbalanced datasets.

The main contributions of this article are summarized as follows:

- We introduce a novel classification algorithm based on the Dempster-Shafer theory of evidence to address the challenges of imbalanced classification.
- Instead of based our work on distance functions that have weaknesses in the situation of a variety of data densities, we introduce a new similarity measure that makes use of the mass-based dissimilarity measurement[1].

The remaining parts of this paper are structured as follows. Section 2 summarizes preliminaries on the imbalanced classification problem, the mass-based dissimilarity measure, and a brief introduction to the Dempster-Shafer theory of evidence. In Section 3, we propose a new EMass method for classifying the imbalanced data sets. First, the confidence of each data point that belongs to its class is estimated. Second, the similarity between the considering instance and its neighbor is evaluated using an estimated dissimilarity measure. Third, each neighboring data point is treated as an independent information source, contributing evidence to predict the target variable. The Dempster rule of combination is then applied to integrate all multiple sources of evidence for final decision making. Section 4 presents and analyzes the results obtained from experiments conducted on 60 imbalanced data sets. Lastly, Section 5 summarizes the conclusions and provides recommendations for future research directions. The notation used in this study is clarified in the following.

- AUC: Area Under The Curve
- BPA: Basic Probability Assignment
- DST: Dempster-Shafer Theory of evidence
- EMass: Evidential Mass-based approach
- k-LMN: k Lowest Mass Neighbors
- SVM: Support Vector Machine

II. Problem Statement and Literature Review

We systematically specify the imbalanced classification problem under the definition of a classification task in Section II.1. The literature review is briefly presented in Section II.2. This covers the dissimilarity measures based on mass approach, and the Dempster-Shafer theory of evidence.

[1]The term *mass* used in this paper in the sense of mass-based dissimilarity introduced in [25]. It is not related to the concept of mass functions used in the Dempster-Shafer theory, which will be referred to as basic probability assignments in the present paper.

1. Imbalanced classification problem statement

In general, classification is a supervised learning task in machine learning, data mining, and knowledge discovery, where the purpose is to predict the target variable of unseen instances based on patterns learned from independent variables. Given the labeled dataset $X = (x, y)$ including n instances $(x_i, y_i), x_i \in \mathbb{R}^d, y_i \in \Omega, 1 \leq i \leq n$, d is the number of dimensions, where $\Omega = \{l_1, l_2, ..., l_M\}$ is the set of M labels, or the frame of discernment.

The goal of classification in predictive modeling is to determine the most probable label $l_i \in \Omega$ for a new unlabeled instance x_t. Almost all classification approaches are based on measuring the similarity between x_t and any or a subset of instances within the dataset, making the definition of similarity a crucial aspect of this research field. However, classification becomes significantly more challenging in the presence of imbalanced datasets, where certain classes dominate others in terms of the number of instances. This imbalance distorts similarity assessments between heterogeneous classes, rendering traditional similarity measures ineffective in such scenarios.

2. Related work

a) Dissimilarity measures based on Mass

Distance functions such as the Manhattan distance and the Euclidean distance are generally accepted to measure the dissimilarity between data points. As a result, numerous distance-based classifiers have been proposed and utilized in a variety of applications. However, these data-independent measurements have shortcomings due to data-dependent properties and assumptions on the distance measures.

In this part, we thoroughly review the measurement of dissimilarity based on mass [25] that rectifies the distance- or density-based algorithms. A mass-based measure may directly replace distance functions, and the remaining parts of the previous algorithms are maintained. It is of interest to note that calculations based on mass inherit their properties from psychological studies in which two instances in a sparse region of space are more likely to be of interest to each other than two instances with the same distance between points in a dense region [26, 27]. For example, most neighbor-based classifiers, such as the k-NN model, face difficulty classifying instances based on the distance measurement, especially when data sets have different data point densities. This problem can be overcome by using mass-based dissimilarity functions. Figure 1 shows the results of the dissimilarity based on the mass values, referring to equation (2). In this example, we selected a hierarchical structure for the dataset X that includes six instances. Then, the mass-based dissimilarity values can be obtained as shown in Figure 1.

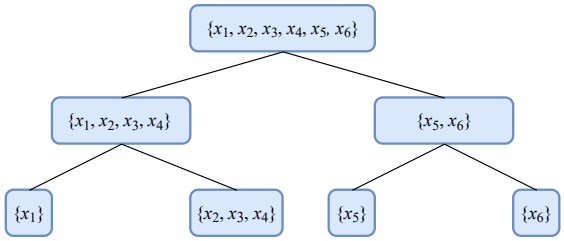


Figure 1. Mass-based dissimilarity measures:

$$\begin{aligned}
&\text{mass}_e(x_1, x_1) = \text{mass}_e(x_5, x_5) = \text{mass}_e(x_6, x_6) = 1/6\\
&\text{mass}_e(x_2, x_3) = \text{mass}_e(x_3, x_4) = \text{mass}_e(x_2, x_4) = 3/6\\
&\text{mass}_e(x_1, x_2) = \text{mass}_e(x_1, x_3) = \text{mass}_e(x_1, x_4) = 4/6\\
&\text{mass}_e(x_2, x_3) = \text{mass}_e(x_2, x_4) = \text{mass}_e(x_3, x_4) = 4/6\\
&\text{mass}_e(x_1, x_5) = \text{mass}_e(x_1, x_6) = \text{mass}_e(x_2, x_5) = 6/6\\
&\text{mass}_e(x_2, x_6) = \text{mass}_e(x_3, x_5) = \text{mass}_e(x_3, x_6) = 6/6
\end{aligned}$$

The following definitions outline the two key steps involved in studying and implementing the mass-based dissimilarity calculation introduced in our previous study [28]. The proposed approach applies the sampling technique to avoid missing any distribution in the dataset. Let H_j be a partition structure of the dataset. H_j contains nested nodes, in which the data points in the same node are more similar than the data points in the different nodes.

Definition 1: Let $S(x_t, x_i|H_j)$ denote the smallest node that contains two instances x_t, x_i with respect to the hierarchical structure H_j, defined by:

$$S(x_t, x_i|H_j) := \underset{S \in H_j, \{x_t, x_i\} \subseteq S}{\text{argmax}} \text{dep}(S|H_j) \tag{1}$$

where $\text{dep}(S|H_j)$ denotes the depth required to separate node S from the initial node within H_j.

Definition 2: Let $\text{mass}(x_t, x_i|H_j)$ denote the dissimilarity function between x_t and x_i, depending on H_j, defined by:

$$\text{mass}(x_t, x_i|H_j) := E_{\mathcal{H}}[P(p|p \in S(x_t, x_i|H_j))]$$

i.e., the expected value of the chance that a random data point $p \in X$ applies to the smallest node $S(x_t, x_i|H_j)$, which is calculated by equation (1) over all possible H_j, where all H_j is denoted by $\mathcal{H}$.

Practically, we select h hierarchical partitioning structures H_j for a given data set X. Therefore, the $\text{mass}(x_t, x_i|H_j)$ can be estimated as the mean of the cardinality of node $S(x_t, x_i|H_j)$ over the set $\mathcal{H}$.

$$\text{mass}_e(x_t, x_i) = \frac{1}{h} \sum_{j=1}^{h} \frac{|S(x_t, x_i|H_j)|}{|H_j|} \tag{2}$$

Definition 3: Let k-LMN(x_t) denote the set of the k-lowest mass-based dissimilarity neighbors around the query instance x_t, defined by:

$$k\text{-LMN}(x_t) = \{x'_1, x'_2, ..., x'_k\}, \quad k \leq |X| \quad \text{(cardinality of } X\text{)} \tag{3}$$

where $x'_i = \underset{x \in X \setminus \{x'_1, x'_2, ..., x'_{i-1}\}}{\text{argmin}} \text{mass}_e(x_t, x), \quad x'_i \in X, \quad i = 1, \ldots, k.$

b) Dempster-Shafer theory (DST)

The theory of the belief function, also known as the evidence theory or DST [29, 30], provides a suitable framework for presenting and reasoning with incomplete information.

Let Ω be a *framework for discernment*, and let 2^Ω be its set of powers. In DST, a basic probability assignment (BPA) concept is defined as a function $m : 2^\Omega \rightarrow [0, 1]$ that satisfies the conditions below:

$$\sum_{A \subseteq \Omega} m(A) = 1 \quad \text{and} \quad m(\emptyset) = 0 \tag{4}$$

where quantity $m(A)$ is a measure of the belief committed exactly to A, supported by multiple sources of evidence. Individual subset $A \subseteq \Omega$ such that $m(A) > 0$ is named a focal element of the m function. The total ignorance degree is represented by $m(\Omega)$.

A useful and important operation in DST is the Dempster rule of combination [30], which is used as a powerful framework to combine multiple sources of evidence in many applications [31]. Consider 2-pieces of evidence, on the same Ω, represented by two BPAs $m_1(.)$ and $m_2(.)$. Dempster's rule of combination is applied to combine them to achieve a new BPA, denoted by $(m_1 \oplus m_2)$ (also called the orthogonal sum of m_1 and m_2), which is defined by:

$$(m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} \quad \text{for } A \neq \emptyset \tag{5}$$

Note that Dempster's rule of combination is only applicable to BPAs m_1 and m_2 that verify the condition $\sum_{B \cap C = \emptyset} m_1(B) m_2(C) < 1$.

III. Proposed Evidential Mass-based Approach: EMass

Distance- and density-based classifiers, such as k-Nearest Neighbors (k-NN), Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), Bagging and AdaBoost, struggle with challenges posed by skewed class distributions. These methods treat all instances equally, despite the fact that the majority class overwhelmingly dominates the dataset, leading to biased classification performance. Then, misclassification could happen due to the neighbors pick-up that does not contribute to the imbalanced classification problem. Misclassification occurs when all classes have the same error rate or probability of being misclassified. In this section, we will present our proposed so-called evidential mass-based approach (EMass for short) to handle this misclassification error. The EMass approach is graphically summarized in Figure 2 and detailed in the following subsections. Notice that the query x_t presents an unknown instance for evaluating the model.

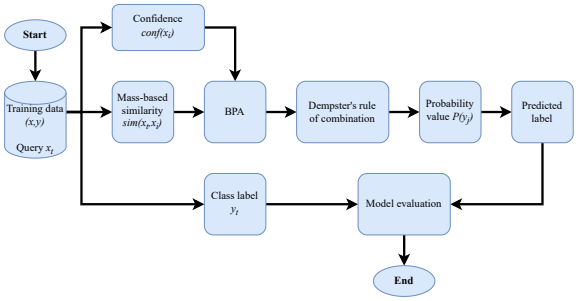


Figure 2. EMass flowchart

1. Symbols and notations

$X = \{x_i, y_i\}$	$\triangleq$	Labeled dataset of n instances
$\Omega = \{l_1, l_2, ..., l_M\}$	$\triangleq$	Set of M labels
mass()	$\triangleq$	Mass-based dissimilarity function
conf()	$\triangleq$	Confidence function
sim()	$\triangleq$	Similarity function
$w()$	$\triangleq$	Weighting function

2. Confidence estimation

We first define the confidence of an instance x_i, denoted by conf(x_i), that provides the degree of confidence in the belonging of the instance x_i to class label y_i. This conf(x_i) is calculated by equation (6), according to Bayes' theorem.

$$\text{conf}(x_i) = P(y_i|x_i) = \frac{P(y_i) \times P(x_i|y_i)}{\sum_{j=1}^{n} P(y_j) \times P(x_i|y_j)} \tag{6}$$

where n is the number of instances, $P(y_i)$ represents the prior probability of class label y_i, and $P(x_i|y_i)$ is the likelihood probability.

The process of estimating the confidence of a data instance is summarized in Algorithm 1 below.

Algorithm 1 *train* Model(x, y)

Input: learning data (x, y)
Output: conf, $\text{mass}_{\max}$

```
conf initialization
mass_max ← 0
for i = 1 to n do
    conf[i] ← compute the confidence, // see the equation (6)
    for j = i + 1 to n do
        mass ← mass_e(x_i, x_j), // refer to [28]
        mass_max ← max(mass, mass_max)
    end for
end for
return conf, mass_max
```

3. Similarity measures based on mass

To define the similarity among data instances, we apply dissimilarity measures based on the mass, as introduced in [28]. The similarity between two data instances should be maximum when they belong to the same leaf node in the hierarchical structure. In contrast, it should be minimal when these instances are in the root node of the partition structure. Formally, we define the similarity between any $x_i, x_t \in X$ as follows:

$$\text{sim}(x_i, x_t) = 1 - \frac{\text{mass}_e(x_i, x_t)}{\text{mass}_{\max}} \tag{7}$$

where $\text{mass}_e(x_i, x_t)$ is the dissimilarity measure based on the mass between two instances as introduced in [28] and $\text{mass}_{\max} = \max_{1 \le i,j \le n}\{\text{mass}_e(x_i, x_j)\}$ denotes the maximum value of all the estimated $\text{mass}_e(x_i, x_j)$.

Those classification methods, such as the NB classifier and the Bayesian belief networks, compute the conditional probability to classify a new instance. However, these methods might not perform accurately due to the skewed distribution of the classes. In addition, the EMass approach calculates the likelihood of an event in which the neighbors of the query instance belong to its context set. In addition, as an uncertain dataset contains noise that makes it deviate from the original values, confidence estimation plays a significant role in addressing imbalanced classification problems where the minority class lacks information. The Dempster rule is applied to merge the multiple sources of evidence contributed by each neighbor of the considered data point to address this limitation.

First, it should be noted that a piece of evidence will allocate a larger BPA value for the query instance to a specific class when its neighbor has a higher confidence or a higher degree of belief that it belongs to the considering class. In other words, evidence with a higher posterior chance will have more belief than evidence with a lower posterior probability value. Second, a piece of evidence will assign more BPA values or a higher degree of belief, for the query instance, to a specific class when the query instance and its neighbor have a higher similarity. We now define the production that satisfies the two mentions above in the following equation.

$$w(x_i, x_t) = \text{conf}(x_i) \times \text{sim}(x_i, x_t) \tag{8}$$

where $\text{conf}(x_i)$ is the confidence of instance x_i estimated by the posterior chance of class label y_i given x_i, and $\text{sim}(x_i, x_t)$ means the similarity between x_i and x_t.

For an unseen query data point x_t, a context set k-LMN(x_t) consists of k-neighbors around x_t having the lowest mass-based dissimilarities measurement. Each instance in k-LMN(x_t) is considered an evidence source that provides a piece of information, which is represented by a BPA defined on the set Ω as in the next subsection. Such evidence supports the prediction of the class label of x_t.

4. Basic probability assignment and evidence combination

We query each i-th neighbor x_i of the considering instance x_t as an evidence source that provides information supporting the hypothesis that the class label of x_t is the same as the class label of x_i, namely, $l_{j(i)} \in \Omega$. Then, based on the observations mentioned above, we formulate the BPA representing the piece of evidence associated with x_i, denoted by m_i, as follows:

$$m_i(\{l_{j(i)}\}) = \beta \times w(x_i, x_t), \tag{9}$$

$$m_i(\Omega) = 1 - \beta \times w(x_i, x_t) \tag{10}$$

$$m_i(A) = 0, \quad \forall A \in 2^{\Omega} \setminus \{\Omega, \{l_{j(i)}\}\} \tag{11}$$

where β is a hyperparameter and $0 < \beta < 1$.

Once the BPA function for the neighbors k $x_i \in k$-LMN(x_t) is defined, Dempster's rule is then applied to combine these k BPA m_i to obtain the overall BPA, denoted by m_t, to determine the label of x_t as follows

$$m_t(.) = \bigoplus_{1 \le i \le k} m_i(.)$$

In particular, we have

$$m_t(\{l'_j\}) = \bigoplus_{1 \le i \le k} m_i(l'_j), \quad l'_j \in \Omega \tag{12}$$

5. Decision making

To make decisions on the label of x_t, the so-called pignistic transformation [32] is applied to transform the combined BPA m_t into the probability distribution as done in [33, 34] as follows:

$$P_t(l'_j) = m_t(\{l'_j\}) + \frac{m_t(\{\Omega\})}{|\Omega|}, \quad l'_j \in \Omega \tag{13}$$

Based on this probability, we make the final decision to predict the class label using equation (14).

$$\hat{y_t} = \underset{l'_j \in \Omega}{\operatorname{argmax}} P_t(l'_j) \tag{14}$$

In summary, the pseudocode of the proposed method is presented in Algorithm 2.

Algorithm 2 EMass approach

Input: learning data (x, y), neighbor number k, considering instance x_t

Output: target variable $\hat{y_t}$

conf, $\text{mass}_{\max} \leftarrow$ learned Model(x, y), // refer to Algorithm 1
$s \leftarrow$ indices of k-LMN(x_t)
Weighted values initialization
for *i = 1 to k* **do**
  $index \leftarrow s[i]$
  conf $\leftarrow$ conf$[index]$
  mass $\leftarrow$ $\text{mass}_e(x_t, x_{index})$, // refer equation (2)
  sim $\leftarrow$ $\text{sim}(x_t, x_{index})$, // refer equation (7)
  BPA$[x_i] \leftarrow$ using equations (9–11)
end for
BPA values combination using equation (12)
Pignistic transformation $\leftarrow$ compute probability values, // refer equation (13)
$\hat{y_t} \leftarrow$ predict class label, // refer equation (14)
Return target label $\hat{y_t}$

IV. Experimental results and discussion

To evaluate our approach in comparison with previous studies, we conducted experiments on 60 imbalanced datasets from the UCI Machine Learning Repository (available at https://archive.ics.uci.edu/). The F1 score and AUC values were used as key evaluation metrics. The experimental results were then subjected to statistical analysis to validate the effectiveness of the EMass model.

1. Experimental datasets

Table I summarizes the characteristics of the 60 imbalanced data sets that were pre-processed for the imbalanced classification task. These datasets were gathered from the UCI machine learning repository [35] and the knowledge extraction process based on the evolutionary learning (KEEL) source [36]. The 60 datasets were chosen from a variety of domains to conduct class imbalance experiments. The numbers of instances, features, and imbalance ratios are all in a wide range of values. The imbalance ratio (IR) of each dataset is defined as the proportion of positive instances to the numbers of negative instances. In this study, the IR spreads from 1.82 up to 100.14.

The original classification datasets were used to generate the unbalanced datasets, as shown in Table I. These data sets were initially collected for conventional binary or multiclass classification tasks. To address the class imbalance problem, the data sets were pre-processed by designating one of the minority classes as the positive class (class 1), while the remaining classes were grouped to form the negative class (class 0). There are data sets in the group that are different versions of each other. However, we consider the positive class to be different from the same original dataset. Therefore, the dependence between data sets did not influence the conclusions or the statistical analysis.

The "Glass" recognition dataset was first experimented with 12 competitively imbalanced classifiers. The investigators used the glass sample, which had nine attributes, as evidence to recognize the type of glass. In total, the data set contains 214 instances distributed across all types of glass. To create the data set "Glass1", glass type 1 was designated as a positive class, while the remaining types were grouped as a negative class. Similarly, we prepared data sets "Glass0", "Glass4", "Glass6", and "Glass-0-4_vs_5" using the same approach.

The Wisconsin Breast Cancer Data Set (abbreviated Wisconsin) consists of 683 medical diagnoses, each described by nine characteristics. In this dataset, the malignant class was designated as a positive class, while the benign class was treated as a negative class. The data set has an imbalance ratio (IR) of 1.86.

The Pima data set was originally collected by the US National Institute of Diabetes, Digestive, and Kidney Diseases to predict diabetes in patients based on eight characteristics. All participants were 21-year-old females. The dataset contains 768 samples, with 268 positive instances, resulting in an imbalance ratio (IR) of 1.87.

The Vehicle dataset, also known as Vehicle Silhouettes [37], was originally designed to classify silhouettes into one of four vehicle types based on a set of characteristics extracted using the Hierarchical Image Processing System (HIPS). This system computes a combination of scale-independent attributes, including scaled variance and skewness/kurtosis along the major and minor axes as classical

moment-based measures. In addition, it employs heuristic measures such as hollows, rectangularity, circularity, and compactness to improve classification accuracy. In this study, we evaluated all 12 competitive approaches in the Vehicle 1 and Vehicle 2 datasets.

The Newthyroid data set categorizes patients into three classes: normal, subnormal, and hyperfunction, using 15 categorical and six numerical features to diagnose potential hypothyroidism. For this experiment, the hyper-function and subnormal classes were grouped as the positive class, while the normal class was designated as the negative class, using five numerical features for classification.

The Segment dataset, also known as the Image Segmentation Data Set, was developed by the University of Massachusetts. It consists of data samples randomly selected from seven outdoor images, where each image was manually segmented at the pixel level to create a classification dataset. Each sample represents a 3×3 pixel region. For this class imbalance experiment, we use Segment0, which contains 2,308 instances, each described by 19 features.

The Yeast data set [38] contains information about yeast cells, originally designed to predict the site of protein localization among 10 possible categories. For this class imbalance experiment, positive and negative classes were defined by selecting appropriate classes from the 10 original categories based on the imbalance ratio (IR) values. For example, in the "Yeas-2_vs_4" dataset, class 2 is designated as the positive class, while class 4 serves as the negative class.

The Page-Blocks dataset consists of document layout blocks identified through a segmentation procedure. It contains 5,472 samples extracted from 54 different documents, each sample representing a single block. The original classification task aims to determine the block type, which falls into one of the following categories: text (0), horizontal line (1), graphic (2), vertical line (3), or picture (4). For this experiment, we prepared the data sets "Page-blocks0" and "Page-blocks-1-3_vs_4" to compare the performance of 12 classifiers.

The Led7digit data set represents a 7-segment display using 7 Boolean features, corresponding to 10 classes, each representing a decimal digit (0-9). The original task was to identify which digit is displayed. Since noise is introduced in these datasets, a pre-processing step is required to remove the noise** before classification. For this experiment, class 1 (representing the digit "1") was designated as the negative class, while the remaining nine classes (0, 2-9) were grouped as the positive class.

The Shuttle datasets share the same nine features but vary in the number of instances. Several versions of the data set were prepared for this experiment. "Shuttle-0_vs_4" contains 1,829 instances, where class 0 is designated as the positive class and class 4 as the negative class; "Shuttle-2_vs_4" includes 129 instances, with class 2 as the positive class and class 4 as the negative class; "Shuttle-6_vs_2-3" consists of 230 instances, where class 6 forms the positive class, while classes 2 and 3 are grouped as the negative class. All remaining classes were removed; "Shuttle-2_vs_5" contains 3,316 instances, with class 2 as the positive class and class 5 as the negative class.

The Dermatology data set consists of 34 characteristics, including one nominal attribute, while the remaining 33 characteristics are numerical. This data set presents a challenging classification task as different erythematosquamous diseases share common clinical characteristics, such as erythema and scaling, with only subtle variations. Initially, 12 clinical characteristics were used for the preliminary diagnosis. Subsequently, skin samples were examined to assess 22 histopathological features, with their values determined by microscopic analysis.

The Winequality data set characterizes the red and white varieties of Portuguese Vinho Verde wine. Due to privacy restrictions, only physicochemical and sensory attributes are available for analysis. The data set consists of ordered but imbalanced classes, as there are significantly more wines of normal quality compared to excellent or poor quality wines. As a result, class imbalance algorithms can be applied to effectively identify excellent and poor wines.

The Poker data set represents a five-card poker hand, where each card is drawn from a standard 52-card deck. Each card is characterized by two attributes, suit and rank, resulting in a total of 10 predictive features. The data set includes a class attribute that defines the type of poker hand, with the order of the cards significant.

The KDDCup99 dataset consists of 34 numerical attributes and 7 categorical attributes. For simplicity, these 41 features were reduced to four: service, duration, src-bytes, and dst-bytes. The data set was then categorized into subsets based on the **service attribute, including HTTP, SMTP, FTP, FTP data, and other protocols.

The Abalone data set contains physical measurements that are used to estimate the age of abalone. In this study, we applied machine learning algorithms to predict the age of new abalone samples. For the class imbalance experiment, each individual class was considered as a positive class with slightly varied proportions.

2. Implementation and evaluation metrics

We compared the EMass approach with the other 11 classifiers, including traditional learning algorithms (C4.5 DT, NB, *k*-NN), logistic regression (LR), tree-based algorithms

Table I
DESCRIPTIONS OF THE IMBALANCED DATASETS. **Idx.**, **#Inst.**, **#Ftr.**, AND **IR** REPRESENT INDEX OF DATASET, NUMBER OF INSTANCES, FEATURES, AND IMBALANCE RATE RESPECTIVELY.

Idx.	Dataset	#Inst.	#Ftr.	IR	Idx.	Dataset	#Inst.	#Ftr.	IR
1	Glass1	214	9	1.82	31	Glass-0-4_vs_5	92	9	10.50
2	Wisconsin	683	9	1.86	32	Ecoli-0-6-7_vs_5	220	6	10.58
3	Pima	768	8	1.87	33	Ecoli-0-1-4-7_vs_2-3-5-6	336	7	11.00
4	Ecoli-0_vs_1	220	7	1.89	34	Led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	11.31
5	Iris0	150	4	2.06	35	Ecoli-0-1_vs_5	240	6	11.63
6	Glass0	214	9	2.10	36	Glass-0-6_vs_5	108	9	12.50
7	Vehicle2	846	18	2.88	37	Ecoli-0-1-4-7_vs_5-6	332	6	12.83
8	Vehicle1	846	18	2.90	38	Ecoli-0-1-4-6_vs_5	280	6	13.74
9	Glass-0-1-2-3_vs_4-5-6	214	9	3.20	39	Shuttle-c0-vs-c4	1829	9	13.87
10	Vehicle0	846	18	3.25	40	Page-blocks-1-3_vs_4	472	10	16.52
11	Ecoli1	336	7	3.36	41	Glass4	214	9	16.83
12	Newthyroid1	215	5	5.14	42	Dermatology-6	358	34	16.90
13	Newthyroid2	215	5	5.32	43	Winequality-white-9_vs_4	168	11	17.67
14	Ecoli2	336	7	5.46	44	Ecoli4	336	7	17.68
15	Segment0	2308	19	6.02	45	Zoo-3	101	16	19.20
16	Glass6	214	9	6.38	46	Poker-9_vs_7	244	10	19.50
17	Yeast3	1484	8	8.10	47	Shuttle-2_vs_4	129	9	20.50
18	Ecoli3	336	7	8.60	48	Shuttle-6_vs_2-3	230	9	22.00
19	Page-blocks0	5472	10	8.79	49	Yeast-2_vs_8	482	8	23.10
20	Yeast-0-2-5-6_vs_3-7-8-9	1004	8	9.14	50	Kddcup-guess_passwd_vs_satan	1642	38	29.98
21	Yeast-0-2-5-7-9_vs_3-6-8	1004	8	9.14	51	Abalone-3_vs_11	502	7	32.47
22	Yeast-2_vs_4	514	8	9.28	52	Yeast5	1484	8	32.73
23	Ecoli-0-6-7_vs_3-5	222	7	9.57	53	Ecoli-0-1-3-7_vs_2-6	281	7	39.42
24	Ecoli-0-1_vs_2-3-5	244	7	9.61	54	Abalone-21_vs_8	581	7	40.50
25	Ecoli-0-2-3-4_vs_5	202	7	9.63	55	Kddcup-land_vs_portsweep	1061	38	49.52
26	Ecoli-0-2-6-7_vs_3-5	224	7	9.67	56	Poker-8-9_vs_6	1485	10	58.40
27	Ecoli-0-4-6_vs_5	203	6	9.68	57	Shuttle-2_vs_5	3316	9	66.67
28	Ecoli-0-3-4-7_vs_5-6	257	7	9.71	58	Kddcup-buffer_overflow_vs_back	2233	38	73.43
29	Ecoli-0-3-4-6_vs_5	205	7	9.79	59	Kddcup-land_vs_satan	1610	38	75.67
30	Vowel0	988	10	9.98	60	Kddcup-rootkit-imap_vs_back	2225	38	100.14

for class imbalance (RF), linear support vector machine (Linear_SVM), SVM with RBF kernel (RBF_SVM), ensemble learning (Bagging, AdaBoost, and XGBoost), and the evidential algorithm (mPE*k*NN).

There are many aspects and methods that can be considered when evaluating the performance of a system for class imbalance problems, such as memory, consumption time, accuracy, F-series score, Brier score, and AUC values. However, for the application of our approach, the AUC results are considered the most important factors to assess the performance when comparing those models. Moreover, the F1 score and Brier score of all testing models are included. For all experiments, we employ a 10-fold cross-validation approach across various methods and datasets. The results presented represent the average results obtained from these validations.

In addition, most imbalance classifiers often demonstrate binary-class problems as multiclass problems, which can be simplified to binary problems. In general, the positive class is considered the minority class, and the negative class is labeled the priority class. Thus, the results of a binary class problem are shown by a confusion matrix. Then, this confusion matrix was used to calculate the F1 score, Brier score, ROC-AUC, and PR-AUC values. Tenfold cross-validation is mainly applied to estimate the skill of all 12 competitive models on the new data. Then, all classifiers are ranked in each dataset in the chosen evaluation metrics.

3. Results and Discussions

Figure 3 a illustrates the comparison of the F1 score in 12 competitive models, evaluated using a tenfold cross-validation in 60 unbalanced datasets. In general, the EMass approach demonstrated superior performance, achieving the highest average rank (7.100) and the best average F1 score (0.857). A detailed comparison of F1 scores is provided in Table A1.

Figure 3b presents a comparison of the average Brier score across 12 models, evaluated using a tenfold cross-validation on 60 unbalanced datasets. The proposed approach outperformed all other models, achieving the highest average rank (9.067) and the lowest average Brier score (0.034), indicating superior performance. It is important to note that a lower Brier score signifies better predictive accuracy. Additionally, the *k*-NN model also achieved competitive results in the Brier score evaluation. Detailed comparisons of Brier scores are provided in Table A2.

Figure 3c illustrates the ROC-AUC comparison among 12 models, evaluated on 60 imbalanced datasets. The results indicate that the EMass model achieved the highest average ROC-AUC value (0.959), while the RBF-SVM model

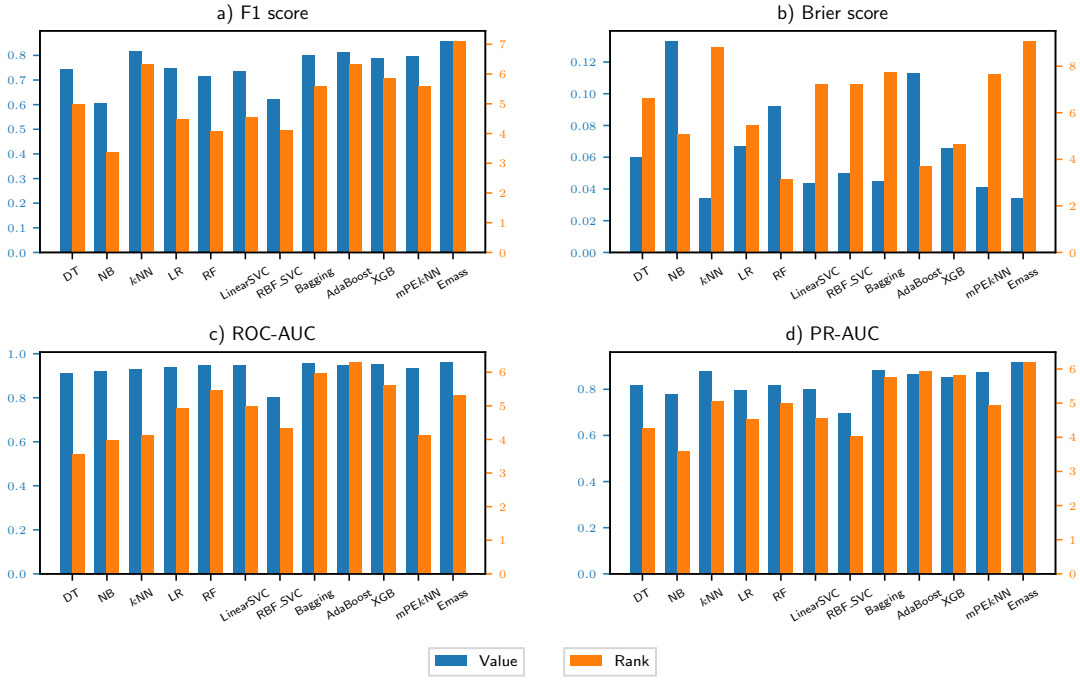


Figure 3. Plots of results comparisons for 12 tested models on 60 imbalanced datasets. Note that a higher average rank indicates better performance.

recorded the lowest average ROC-AUC value (0.800). In terms of average rankings, AdaBoost obtained the best performance (6.283), outperforming all other models. Detailed ROC-AUC measurements are provided in Table A3.

Figure 3d presents the average PR-AUC comparison between 12 models, evaluated using ten-fold cross-validation on the same datasets. In general, the EMass approach outperformed all competing models, achieving the highest mean PR-AUC value (0.915) and the best average ranking (6.183) in Table A4 reports a detailed comparison of PR-AUC results.

4. Wilcoxon signed ranks test results

We used the Wilcoxon signed ranks test [39] as a nonparametric statistical analysis to experiment with the pairwise comparison in terms of the F1 score, Brier score and AUC results. This analysis carries out multiple pairwise comparisons between the EMass approach and each of the other methods. The SPSS software package was used on all the experimental results and the output is presented in Table II. Note that $\mathbf{R}^+$ is defined as the sum of positive signed ranks and $\mathbf{R}^-$ as the sum of negative signed ranks for the results of each comparison method.

It can be seen in Table II that the RBF_SVM model achieved the best results $\mathbf{R}^-$ in terms of the F1 score (**1362.5**) and the PR-AUC metric (**883.0**) compared to the other methods. However, the proposed approach outperforms this model according to the multiple method comparison results presented in Table A1 and Table A4 There are significant differences between EMass and RBF_SVM with a confidence level higher than 99.9% (p-value = 0.001).

In terms of the Brier score, k-NN is the best algorithm from pairwise comparison with EMass due to the highest value $\mathbf{R}^-$ (**758.0**). However, the EMass approach outperforms it according to the results of the comparison of multiple methods in Table A2. However, RF is the worst model among the other methods. In the ROC-AUC metric, the DT model obtained the best $\mathbf{R}^-$ value (**772.0**) compared to the other algorithms, and the test reports a p-value = 0.001. This means that the confidence level is higher than 99.9% for the pairwise method comparison result.

V. Conclusion

In this study, we address the imbalanced classification problem through the lens of neighbor-based algorithms, mass-based similarity measurements, and evidential reasoning methods. Our research highlights that misclassification errors can be mitigated by considering each neighbor as a source of evidence, rather than relying solely on the query instance. To overcome the limitations of distance-based and density-based classifiers, we introduce a mass-based model to measure the similarity between instances. Additionally, we define instance confidence to mitigate the impact of skewed class distributions, weighting this confidence by

Table II
WILCOXON SIGNED RANKS TEST RESULTS.

EMass	F1 results			Brier results			ROC-AUC results			PR-AUC results		
	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
DT	150.0	840.0	0.001	1075.0	200.0	0.001	174.0	**772.0**	0.001	156.0	790.0	0.001
NB	74.0	1054.0	0.001	1592.0	119.0	0.001	259.5	643.5	0.016	77.0	869.0	0.001
k-NN	256.5	373.5	0.338	727.0	**758.0**	0.894	106.0	455.0	0.002	191.5	438.5	0.043
LR	111.5	923.5	0.001	1572.0	258.0	0.001	349.5	511.5	0.294	152.0	751.0	0.001
RF	119.0	1057.0	0.001	1712.0	58.0	0.001	447.0	499.0	0.754	235.5	754.5	0.002
LinearSVM	120.0	915.0	0.001	1291.0	539.0	0.006	340.0	521.0	0.241	209.0	737.0	0.001
RBF_SVM	122.5	**1362.5**	0.001	1202.0	628.0	0.035	238.0	752.0	0.003	152.0	**883.0**	0.001
Bagging	228.0	592.0	0.014	1058.0	373.0	0.002	399.5	266.5	0.296	247.0	419.0	0.177
AdaBoost	206.5	496.5	0.029	1741.0	89.0	0.001	438.0	265.0	0.192	238.0	465.0	0.087
XGB	284.5	705.5	0.014	1664.0	166.0	0.001	400.5	460.5	0.697	290.0	571.0	0.069
mPEkNN	186.5	633.5	0.003	1027.0	458.0	0.014	114.5	480.5	0.002	184.5	481.5	0.020

mass-based similarity measures between the query instance and its neighbors. Each neighbor is treated as a piece of evidence supporting the label prediction, and Dempster's rule of combination is applied to aggregate these pieces of evidence for final decision-making. As a result, we propose a new classification algorithm for imbalanced data, called EMass.

Traditional classification methods predominantly rely on distance functions, which impose restrictive assumptions such as data independence and adherence to distance axioms that are often unrealistic in real-world datasets. In contrast, our approach leverages mass-based similarity measurements rather than distance-based functions, allowing greater flexibility in handling uncertain and dependent data. EMass follows Dempster-Shafer Theory (DST) to model and integrate multiple pieces of evidence, making it more robust for reasoning under uncertainty in imbalanced datasets.

Our empirical results demonstrate that EMass outperforms 11 competitive models in 60 imbalanced data sets, evaluated using the F1 score, the Brier score, and the AUC values. However, we recognize that the proposed approach is currently limited to datasets with complete numerical variables. Furthermore, the data sets used in our experiments have a restricted number of instances (maximum of 5,472) and features (maximum of 38). In modern classification problems, data sets often contain significantly more instances and features, as well as incomplete and mixed data types. For future research, our objective is to improve mass-based similarity measures, extend the approach to handle incomplete data, and improve adaptability for diverse data types to further increase its applicability in real-world classification tasks.

ACKNOWLEDGMENT

This research is funded by the University of Economics Ho Chi Minh City (UEH), Vietnam. [Grant number: CTD-2024-03]Its results were partially presented at Seminar, School of Business Information Technology, College of Technology and Design, UEH (see: https://bit.ueh.edu.vn/tham-luan-hoi-thao-khoa-hoc-cap-khoa-seminar-nam-2023).

REFERENCES

[1] R. Rastogi and K. Shim, "Public: A decision tree classifier that integrates building and pruning," *Data Mining and Knowledge Discovery*, vol. 4, pp. 315–344, 2000.

[2] J. Hatwell, M. M. Gaber, and R. M. A. Azad, "Ada-WHIPS: explaining AdaBoost classification with applications in the health sciences," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–25, 2020.

[3] F. Thabtah, P. Cowling, and Y. Peng, "Mcar: multi-class classification based on association rule," in *The 3rd ACS/IEEE International Conference on Computer Systems and Applications*. IEEE, 2005, p. 33.

[4] Z. Hao, X. Wang, L. Yao, and Y. Zhang, "Improved classification based on predictive association rules," in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 1165–1170.

[5] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *arXiv preprint arXiv:1302.6815*, 2013.

[6] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for k-NN classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, pp. 1–19, 2017.

[7] H. Cheng, X. Yan, J. Han, and S. Y. Philip, "Direct discriminative pattern mining for effective classification," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 169–178.

[8] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[9] J. Gehrke, R. Ramakrishnan, and V. Ganti, "Rainforest—a framework for fast decision tree construction of large datasets," *Data Mining and Knowledge Discovery*, vol. 4, pp. 127–162, 2000.

[10] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognition Letters*, vol. 136, pp. 190–197, 2020.

[11] M. Ring and B. M. Eskofier, "An approximation of the Gaussian RBF kernel for efficient classification with SVMs," *Pattern Recognition Letters*, vol. 84, pp. 107–113, 2016.

[12] H. Yu, J. Yang, and J. Han, "Classifying large data sets using svms with hierarchical clusters," in *Proceedings of the Ninth*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 306–315.
[13] W. M. Wang, J. Wang, Z. Li, Z. Tian, and E. Tsui, "Multiple affective attribute classification of online customer product reviews: A heuristic deep learning method for supporting kansei engineering," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 33–45, 2019.
[14] Y. Villuendas-Rey, C. F. Rey-Benguría, Á. Ferreira-Santiago, O. Camacho-Nieto, and C. Yáñez-Márquez, "The naïve associative classifier (nac): a novel, simple, transparent, and accurate classification model evaluated on financial data," *Neurocomputing*, vol. 265, pp. 105–115, 2017.
[15] Y. Villuendas-Rey, M. D. Alanis-Tamez, C. F. R. Benguría, C. Yáñez-Márquez, and O. Camacho-Nieto, "Medical diagnosis of chronic diseases based on a novel computational intelligence algorithm," *Journal of Universal Computer Science*, pp. 775–796, 2018.
[16] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets," *Soft Computing*, vol. 13, pp. 213–225, 2009.
[17] Y. SUN, A. K. C. WONG, and M. S. KAMEL, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.
[18] A. G. de Sá, A. C. Pereira, and G. L. Pappa, "A customized classification algorithm for credit card fraud detection," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 21–29, 2018.
[19] P. Tapkan, L. Özbakır, S. Kulluk, and A. Baykasoğlu, "A cost-sensitive classification algorithm: BEE-Miner," *Knowledge-Based Systems*, vol. 95, pp. 99–113, 2016.
[20] D. Gan, J. Shen, B. An, M. Xu, and N. Liu, "Integrating TANBN with cost sensitive classification algorithm for imbalanced data in medical diagnosis," *Computers & Industrial Engineering*, vol. 140, p. 106266, 2020.
[21] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, and H. Fujita, "Multi-imbalance: An open-source software for multi-class imbalance learning," *Knowledge-Based Systems*, vol. 174, pp. 137–143, 2019.
[22] Y. Pan, L. Zhang, X. Wu, and M. J. Skibniewski, "Multi-classifier information fusion in risk analysis," *Information Fusion*, vol. 60, pp. 121–136, 2020.
[23] C. Zhu, B. Qin, F. Xiao, Z. Cao, and H. M. Pandey, "A fuzzy preference-based Dempster-Shafer evidence theory for decision fusion," *Information Sciences*, vol. 570, pp. 306–322, 2021.
[24] X. Xu, D. Zhang, Y. Bai, L. Chang, and J. Li, "Evidence reasoning rule-based classifier with uncertainty quantification," *Information Sciences*, vol. 516, pp. 192–204, 2020.
[25] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Singapore: Springer, 2016, pp. 1205–1214.
[26] A. Tversky, "Features of similarity." *Psychological Review*, vol. 84, no. 4, p. 327, 1977.
[27] C. L. Krumhansl, "Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density," *American Psychologist*, vol. 5, pp. 445–463, 1978.
[28] A. Hoang, T. N. Mau, D.-V. Vo, and V.-N. Huynh, "A mass-based approach for local outlier detection," *IEEE Access*, vol. 9, pp. 16 448–16 466, 2021.
[29] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," in *"Classic Works of the Dempster-Shafer Theory of Belief Functions"*. Berlin Heidelberg: Springer, 2008, pp. 57–72.
[30] G. Shafer, *A Mathematical Theory of Evidence*. New Jersey: Princeton University Press, 1976.
[31] T. N. Mau, Q. Le, D.-V. Vo, D. Doan, and V.-N. Huynh, "Integrating machine learning and evidential reasoning for user profiling and recommendation," *Journal of Systems Science and Systems Engineering*, vol. 32, pp. 393–412, 2023.
[32] P. Smets, "Decision making in the TBM: the necessity of the pignistic transformation," *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 133–147, 2005.
[33] E. Ramasso and R. Gouriveau, "Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions," in *2010 Prognostics and System Health Management Conference*. IEEE, 2010, pp. 1–10.
[34] Z. Su, T. Denoeux, Y. Hao, and M. Zhao, "Evidential k-NN classification with enhanced performance via optimizing a class of parametric conjunctive t-rules," *Knowledge-Based Systems*, vol. 142, pp. 7–16, 2018.
[35] M. Lichman *et al.*, "UCI Machine Learning Repository," 2013.
[36] I. Triguero, S. González, J. M. Moyano, S. García, J. Alcalá-Fdez, J. Luengo, A. Fernández, M. J. del Jesús, L. Sánchez, and F. Herrera, "Keel 3.0: An open source software for multi-stage analysis in data mining," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 1238–1249, 2017.
[37] J. P. Siebert, "Vehicle recognition using rule based methods," *Turing Institute Research Memorandum TIRM-87-0.18*, 1987.
[38] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Bioinformatics*, vol. 11, no. 2, pp. 95–110, 1991.
[39] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York: Springer, 1992, pp. 196–202.

Anh Hoang received the B.S. degree in Telecommunication engineer from the Department of Electrical, Electronic, and Information Engineering, Hanoi University of Transport and Communication, in 2007, and the M.S. degree in Computer Science from the National Taiwan University of Science and Technology (NTUST), Taiwan, in 2010. He completed Ph.D. program at the Graduate School of Advanced Science and Technology (major in Knowledge Science), Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2021. He is currently teaching at School of Business Information Technology, College of Technology and Design, University of Economics Ho Chi Minh City (UEH), Vietnam. His research interests are related to AI/Machine Learning, Data Science/Data Mining/Data Analytics, CyberSecurity, and Business Intelligence/ Business Analytics
Email: Anhh@ueh.edu.vn

APPENDIX: DETAILED COMPARISON RESULTS FOR 12 TESTED METHODS ON 60 IMBALANCED DATASETS.

Table A1
F1 SCORES COMPARISON.

Idx.	Dataset	DT	NB	*k*NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPE*k*NN	**EMass**
1	Glass1	0.722	0.591	0.581	0.526	0.692	0.591	0.491	0.688	0.667	0.647	0.650	0.778
2	Wisconsin	0.952	0.962	0.961	0.990	0.944	0.981	0.971	0.953	0.981	0.952	0.981	0.981
3	Pima	0.639	0.684	0.565	0.624	0.656	0.624	0.677	0.703	0.651	0.646	0.639	0.614
4	Ecoli-0_vs_1	0.963	0.783	1.000	1.000	0.963	1.000	1.000	0.963	0.963	0.963	0.963	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.667	0.556	0.545	0.483	0.692	0.516	0.407	0.692	0.741	0.769	0.581	0.688
7	Vehicle2	0.819	0.583	0.833	0.927	0.819	0.911	0.509	0.819	0.953	0.952	0.800	0.962
8	Vehicle1	0.615	0.523	0.483	0.693	0.504	0.710	0.513	0.667	0.585	0.673	0.558	0.647
9	Glass-0-1-2-3_vs_4-5-6	0.846	0.786	0.889	0.897	0.769	0.889	0.517	0.929	1.000	0.889	0.929	1.000
10	Vehicle0	0.822	0.609	0.882	0.946	0.643	0.946	0.622	0.643	0.939	0.909	0.909	0.878
11	Ecoli1	0.733	0.343	0.667	0.690	0.759	0.733	0.714	0.710	0.640	0.769	0.690	0.714
12	New-thyroid1	1.000	0.923	0.909	1.000	1.000	1.000	0.909	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.941	1.000	1.000	0.933	0.941	0.933	0.875	0.941	1.000	0.941	1.000	1.000
14	Ecoli2	0.783	0.339	0.818	0.690	0.741	0.667	0.783	0.692	0.783	0.818	0.750	0.720
15	Segment0	0.970	0.615	0.969	0.992	0.590	0.992	0.558	0.955	0.985	0.962	0.961	0.984
16	Glass6	0.750	0.875	0.857	0.762	0.778	0.889	0.000	0.857	0.857	0.875	0.800	0.857
17	Yeast3	0.850	0.272	0.800	0.694	0.698	0.687	0.764	0.791	0.778	0.840	0.708	0.735
18	Ecoli3	0.636	0.516	0.615	0.556	0.533	0.615	0.667	0.667	0.706	0.769	0.706	0.778
19	Page-blocks0	0.589	0.500	0.805	0.736	0.631	0.688	0.275	0.585	0.813	0.850	0.768	0.822
20	Yeast-0-2-5-7-9_vs_3-6-8	0.741	0.230	0.851	0.656	0.704	0.702	0.741	0.741	0.755	0.667	0.741	0.755
21	Yeast-0-2-5-6_vs_3-7-8-9	0.615	0.313	0.700	0.471	0.582	0.516	0.571	0.615	0.576	0.566	0.595	0.615
22	Yeast-2_vs_4	0.710	0.818	0.842	0.667	0.846	0.692	0.750	0.667	0.880	0.645	0.696	0.900
23	Ecoli-0-6-7_vs_3-5	0.857	0.000	1.000	0.600	0.600	0.667	0.667	1.000	0.800	0.857	0.857	0.800
24	Ecoli-0-1_vs_2-3-5	0.267	0.000	1.000	0.500	0.500	0.444	0.800	0.800	0.571	0.500	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.615	0.444	0.889	0.727	0.667	0.727	0.727	0.889	0.750	0.889	0.889	0.889
26	Ecoli-0-2-6-7_vs_3-5	0.889	0.000	1.000	0.533	0.667	0.667	0.615	0.889	0.800	0.727	0.857	0.857
27	Ecoli-0-4-6_vs_5	0.800	0.750	0.857	0.750	0.857	0.750	0.857	0.667	0.857	0.800	0.857	0.857
28	Ecoli-0-3-4-7_vs_5-6	0.588	0.400	0.909	0.625	0.714	0.769	0.833	0.800	1.000	0.625	0.667	0.769
29	Ecoli-0-3-4-6_vs_5	0.462	0.727	0.889	0.667	0.889	0.727	0.800	0.600	0.800	0.500	0.889	0.889
30	Vowel0	0.875	0.792	1.000	0.724	0.731	0.724	0.828	0.875	0.958	0.875	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	0.667	0.800	1.000	0.800	0.000	1.000	1.000	1.000	1.000	0.667
32	Ecoli-0-6-7_vs_5	0.615	1.000	0.857	0.500	0.800	0.800	0.889	0.889	1.000	1.000	0.800	0.667
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.600	0.444	0.667	0.533	0.533	0.533	0.625	0.667	0.571	0.667	0.500	0.600
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.769	0.667	0.800	0.625	0.647	0.769	0.769	0.769	0.625	0.710	0.783	0.783
35	Ecoli-0-1_vs_5	0.833	0.833	0.923	0.875	1.000	0.857	0.923	0.833	0.833	0.769	0.923	0.923
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.286	0.400	0.000	1.000	1.000	1.000	0.667	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.444	0.750	0.667	0.571	0.400	0.429	0.600	0.444	0.727	0.444	0.667	0.750
38	Ecoli-0-1-4-6_vs_5	0.600	0.889	1.000	0.571	0.889	0.571	0.889	0.857	0.800	0.667	0.889	1.000
39	Shuttle-c0-vs-c4	1.000	0.967	0.983	0.983	1.000	1.000	0.968	1.000	1.000	1.000	0.983	1.000
40	Page-blocks-1-3_vs_4	0.947	0.526	0.800	0.857	0.581	0.783	0.455	0.947	0.947	0.947	0.778	0.941
41	Glass4	0.400	0.000	1.000	0.667	0.667	0.800	0.170	0.750	0.667	0.750	0.800	1.000
42	Dermatology-6	1.000	0.333	1.000	1.000	1.000	1.000	0.444	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.500	0.000	0.000	0.667	0.000	0.667	0.250	0.667	0.000	0.400	0.000	0.667
44	Ecoli4	1.000	0.522	1.000	0.857	0.667	0.800	0.800	1.000	0.833	1.000	0.923	1.000
45	Zoo-3	0.333	0.667	0.000	0.667	0.400	0.667	0.400	0.000	0.667	0.667	1.000	0.667
46	Poker-9_vs_7	0.222	0.667	0.667	0.000	0.333	0.000	0.800	0.667	0.667	0.500	0.500	0.667
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.074	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.800
49	Yeast-2_vs_8	0.169	0.800	0.800	0.800	0.667	0.533	0.615	0.727	0.667	0.727	0.667	0.800
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	0.889	0.889	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.593	0.168	0.875	0.615	0.485	0.471	0.500	0.593	0.632	0.750	0.700	0.609
53	Ecoli-0-1-3-7_vs_2-6	0.286	0.667	1.000	0.286	0.500	0.286	0.333	0.667	0.667	0.000	0.667	0.667
54	Abalone-21_vs_8	0.571	0.381	0.571	0.889	0.348	0.800	0.727	0.800	0.667	0.444	0.500	0.750
55	Kddcup-land_vs_portsweep	1.000	0.010	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.025	0.000	0.000	0.035	0.000	0.020	0.000	0.000	0.030	0.067	0.000	1.000
57	Shuttle-2_vs_5	1.000	0.116	1.000	1.000	1.000	1.000	0.667	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.941	1.000	0.941	0.842	1.000	0.842	1.000	1.000	1.000	0.933	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.667	1.000	1.000	0.667	0.400	1.000	1.000	1.000	0.667	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	0.800	1.000	0.857	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.744	0.605	0.818	0.747	0.716	0.738	0.622	0.801	0.813	0.790	0.796	**0.857**
	Avg. Ranks	5.000	3.383	6.350	4.483	4.100	4.567	4.133	5.600	6.333	5.850	5.583	**7.100**

Table A2
BRIER SCORES COMPARISON.

Idx.	Dataset	DT	NB	*k*-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPE*k*NN	**EMass**
1	Glass1	0.174	0.354	0.171	0.239	0.194	0.201	0.220	0.166	0.229	0.164	0.264	0.172
2	Wisconsin	0.034	0.029	0.025	0.012	0.044	0.016	0.013	0.032	0.163	0.059	0.015	0.015
3	Pima	0.192	0.172	0.240	0.190	0.212	0.173	0.168	0.176	0.239	0.183	0.221	0.241
4	Ecoli-0_vs_1	0.011	0.080	0.013	0.011	0.073	0.007	0.004	0.009	0.100	0.049	0.017	0.001
5	Iris0	0.000	0.000	0.000	0.033	0.001	0.000	0.000	0.000	0.000	0.038	0.000	0.000
6	Glass0	0.172	0.354	0.147	0.186	0.155	0.151	0.193	0.124	0.198	0.122	0.263	0.227
7	Vehicle2	0.087	0.169	0.072	0.042	0.170	0.041	0.194	0.091	0.189	0.055	0.079	0.024
8	Vehicle1	0.155	0.279	0.192	0.126	0.221	0.131	0.185	0.153	0.236	0.140	0.208	0.177
9	Glass-0-1-2-3_vs_4-5-6	0.094	0.120	0.062	0.055	0.097	0.067	0.251	0.056	0.135	0.079	0.046	0.003
10	Vehicle0	0.083	0.299	0.042	0.015	0.165	0.026	0.132	0.123	0.188	0.070	0.050	0.059
11	Ecoli1	0.097	0.657	0.096	0.100	0.126	0.081	0.082	0.090	0.168	0.092	0.131	0.111
12	New-thyroid1	0.002	0.008	0.013	0.003	0.037	0.008	0.047	0.004	0.051	0.040	0.006	0.002
13	Newthyroid2	0.023	0.000	0.005	0.023	0.025	0.012	0.043	0.016	0.030	0.050	0.006	0.000
14	Ecoli2	0.073	0.560	0.057	0.108	0.142	0.087	0.054	0.073	0.177	0.073	0.092	0.103
15	Segment0	0.008	0.161	0.009	0.002	0.115	0.004	0.042	0.013	0.084	0.040	0.010	0.004
16	Glass6	0.100	0.044	0.054	0.073	0.061	0.102	0.153	0.035	0.136	0.055	0.065	0.033
17	Yeast3	0.056	0.610	0.045	0.075	0.144	0.052	0.043	0.055	0.177	0.070	0.084	0.079
18	Ecoli3	0.079	0.208	0.056	0.081	0.127	0.067	0.055	0.070	0.162	0.062	0.072	0.062
19	Page-blocks0	0.075	0.090	0.032	0.067	0.123	0.072	0.074	0.091	0.210	0.056	0.041	0.034
20	Yeast-0-2-5-6_vs_3-7-8-9	0.137	0.091	0.059	0.156	0.183	0.067	0.059	0.118	0.234	0.129	0.069	0.071
21	Yeast-0-2-5-7-9_vs_3-6-8	0.076	0.532	0.033	0.099	0.144	0.038	0.030	0.067	0.216	0.094	0.056	0.055
22	Yeast-2_vs_4	0.062	0.034	0.032	0.092	0.110	0.029	0.027	0.052	0.189	0.076	0.046	0.024
23	Ecoli-0-6-7_vs_3-5	0.052	0.088	0.010	0.121	0.111	0.044	0.013	0.020	0.189	0.059	0.006	0.021
24	Ecoli-0-1_vs_2-3-5	0.104	0.040	0.002	0.070	0.110	0.025	0.005	0.037	0.181	0.071	0.000	0.000
25	Ecoli-0-2-3-4_vs_5	0.104	0.204	0.011	0.082	0.081	0.066	0.029	0.029	0.110	0.064	0.024	0.024
26	Ecoli-0-2-6-7_vs_3-5	0.064	0.108	0.005	0.093	0.119	0.047	0.012	0.027	0.186	0.068	0.020	0.024
27	Ecoli-0-4-6_vs_5	0.024	0.037	0.014	0.070	0.059	0.059	0.020	0.041	0.144	0.062	0.021	0.024
28	Ecoli-0-3-4-7_vs_5-6	0.079	0.240	0.015	0.088	0.117	0.031	0.013	0.036	0.164	0.084	0.022	0.042
29	Ecoli-0-3-4-6_vs_5	0.136	0.054	0.027	0.103	0.104	0.075	0.019	0.052	0.144	0.113	0.027	0.025
30	Vowel0	0.032	0.047	0.000	0.058	0.104	0.038	0.021	0.031	0.129	0.051	0.000	0.000
31	Glass-0-4_vs_5	0.000	0.000	0.023	0.053	0.017	0.008	0.098	0.000	0.000	0.039	0.012	0.026
32	Ecoli-0-6-7_vs_5	0.043	0.017	0.023	0.121	0.083	0.073	0.013	0.025	0.172	0.046	0.030	0.032
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.075	0.074	0.049	0.098	0.119	0.049	0.041	0.055	0.187	0.084	0.057	0.061
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.084	0.081	0.057	0.076	0.122	0.061	0.048	0.067	0.214	0.095	0.048	0.054
35	Ecoli-0-1_vs_5	0.042	0.029	0.021	0.035	0.050	0.093	0.022	0.035	0.087	0.074	0.021	0.021
36	Glass-0-6_vs_5	0.000	0.000	0.005	0.004	0.123	0.027	0.065	0.000	0.000	0.044	0.012	0.000
37	Ecoli-0-1-4-7_vs_5-6	0.073	0.034	0.038	0.069	0.100	0.047	0.039	0.053	0.111	0.078	0.039	0.023
38	Ecoli-0-1-4-6_vs_5	0.056	0.020	0.002	0.097	0.065	0.047	0.005	0.032	0.142	0.066	0.005	0.002
39	Shuttle-c0-vs-c4	0.000	0.005	0.003	0.006	0.009	0.000	0.003	0.000	0.000	0.034	0.003	0.000
40	Page-blocks-1-3_vs_4	0.011	0.086	0.026	0.030	0.095	0.044	0.060	0.009	0.011	0.042	0.026	0.007
41	Glass4	0.070	0.195	0.013	0.086	0.119	0.081	0.088	0.039	0.076	0.065	0.028	0.009
42	Dermatology-6	0.000	0.111	0.003	0.001	0.047	0.004	0.015	0.001	0.000	0.035	0.007	0.000
43	Winequality-white-9_vs_4	0.058	0.058	0.059	0.029	0.064	0.052	0.055	0.042	0.043	0.082	0.059	0.018
44	Ecoli4	0.000	0.155	0.005	0.015	0.086	0.005	0.005	0.006	0.096	0.037	0.015	0.000
45	Zoo-3	0.129	0.048	0.042	0.048	0.115	0.073	0.097	0.071	0.097	0.069	0.022	0.048
46	Poker-9_vs_7	0.116	0.026	0.016	0.140	0.118	0.039	0.024	0.050	0.021	0.071	0.020	0.012
47	Shuttle-c2-vs-c4	0.000	0.000	0.000	0.000	0.004	0.000	0.037	0.000	0.000	0.037	0.000	0.000
48	Shuttle-6_vs_2-3	0.000	0.000	0.000	0.031	0.038	0.003	0.001	0.000	0.000	0.036	0.000	0.009
49	Yeast-2_vs_8	0.178	0.021	0.027	0.246	0.169	0.028	0.026	0.092	0.195	0.076	0.029	0.025
50	Kddcup-guess_passwd_vs_satan	0.000	0.000	0.000	0.000	0.003	0.001	0.000	0.000	0.000	0.034	0.000	0.000
51	Abalone-3_vs_11	0.000	0.000	0.000	0.000	0.000	0.003	0.001	0.000	0.000	0.035	0.000	0.000
52	Yeast5	0.020	0.256	0.010	0.028	0.069	0.012	0.011	0.023	0.092	0.042	0.020	0.029
53	Ecoli-0-1-3-7_vs_2-6	0.053	0.018	0.004	0.075	0.064	0.016	0.014	0.026	0.069	0.049	0.010	0.018
54	Abalone-21_vs_8	0.037	0.093	0.028	0.009	0.096	0.018	0.021	0.040	0.156	0.057	0.031	0.019
55	Kddcup-land_vs_portsweep	0.000	0.906	0.000	0.000	0.003	0.001	0.005	0.000	0.000	0.034	0.000	0.000
56	Poker-8-9_vs_6	0.197	0.017	0.018	0.254	0.210	0.016	0.017	0.145	0.235	0.094	0.019	0.000
57	Shuttle-2_vs_5	0.000	0.070	0.001	0.000	0.005	0.000	0.005	0.000	0.000	0.034	0.001	0.000
58	Kddcup-buffer_overflow_vs_back	0.000	0.002	0.000	0.002	0.019	0.002	0.001	0.000	0.000	0.034	0.001	0.000
59	Kddcup-land_vs_satan	0.000	0.000	0.003	0.001	0.007	0.003	0.003	0.000	0.000	0.034	0.003	0.000
60	Kddcup-rootkit-imap_vs_back	0.000	0.000	0.000	0.001	0.026	0.001	0.000	0.000	0.000	0.034	0.000	0.000
	Avg. Values	0.060	0.133	**0.034**	0.067	0.092	0.044	0.050	0.045	0.113	0.066	0.041	**0.034**
	Avg. Ranks	6.633	5.083	8.817	5.483	3.150	7.233	7.233	7.750	3.717	4.650	7.667	**9.067**

Table A3
ROC-AUC RESULTS COMPARISON.

Idx.	Dataset	DT	NB	*k*-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPE*k*NN	**EMass**
1	Glass1	0.841	0.677	0.813	0.626	0.800	0.709	0.330	0.855	0.791	0.877	0.892	0.961
2	Wisconsin	0.982	0.994	0.987	0.993	0.989	0.993	0.988	0.991	0.997	0.991	0.987	0.987
3	Pima	0.804	0.825	0.701	0.815	0.812	0.799	0.806	0.830	0.813	0.808	0.744	0.722
4	Ecoli-0_vs_1	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.825	0.781	0.832	0.807	0.896	0.815	0.287	0.925	0.940	0.912	0.841	0.895
7	Vehicle2	0.938	0.809	0.936	0.986	0.972	0.986	0.744	0.958	0.995	0.997	0.934	0.978
8	Vehicle1	0.815	0.679	0.740	0.892	0.663	0.891	0.690	0.856	0.836	0.875	0.754	0.823
9	Glass-0-1-2-3_vs_4-5-6	0.867	0.955	0.960	0.981	0.969	0.988	0.024	0.988	1.000	0.929	0.964	1.000
10	Vehicle0	0.947	0.807	0.984	0.998	0.839	0.995	0.857	0.943	0.998	0.993	0.980	0.979
11	Ecoli1	0.883	0.847	0.845	0.934	0.884	0.940	0.945	0.892	0.891	0.890	0.858	0.895
12	New-thyroid1	1.000	1.000	1.000	1.000	1.000	1.000	0.982	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.986	1.000	1.000	1.000	1.000	1.000	0.896	1.000	1.000	1.000	1.000	1.000
14	Ecoli2	0.860	0.861	0.892	0.890	0.915	0.874	0.895	0.902	0.957	0.909	0.889	0.893
15	Segment0	0.990	0.987	0.983	1.000	0.987	1.000	0.971	0.986	0.998	0.990	0.983	0.985
16	Glass6	0.850	0.971	0.864	0.982	0.989	0.986	0.050	0.982	0.975	0.993	0.864	0.986
17	Yeast3	0.947	0.966	0.938	0.968	0.980	0.962	0.977	0.973	0.967	0.973	0.935	0.972
18	Ecoli3	0.916	0.950	0.858	0.927	0.965	0.935	0.979	0.949	0.950	0.975	0.867	0.918
19	Page-blocks0	0.954	0.931	0.940	0.973	0.938	0.960	0.880	0.955	0.988	0.985	0.944	0.942
20	Yeast-0-2-5-6_vs_3-7-8-9	0.843	0.851	0.869	0.853	0.899	0.854	0.832	0.867	0.843	0.879	0.862	0.842
21	Yeast-0-2-5-7-9_vs_3-6-8	0.959	0.954	0.985	0.960	0.950	0.961	0.969	0.973	0.960	0.959	0.981	0.983
22	Yeast-2_vs_4	0.981	0.980	0.900	0.963	0.993	0.975	0.988	0.994	0.998	0.986	0.897	0.904
23	Ecoli-0-6-7_vs_3-5	0.988	0.976	1.000	0.976	0.984	0.976	1.000	1.000	1.000	0.992	1.000	1.000
24	Ecoli-0-1_vs_2-3-5	0.888	1.000	1.000	0.979	1.000	0.989	1.000	1.000	1.000	0.936	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.973	1.000	1.000	0.953	0.986	0.959	1.000	1.000	0.993	1.000	1.000	1.000
26	Ecoli-0-2-6-7_vs_3-5	1.000	0.951	1.000	0.976	1.000	0.970	1.000	1.000	0.976	0.994	1.000	1.000
27	Ecoli-0-4-6_vs_5	0.833	0.991	1.000	0.982	0.991	0.982	1.000	0.982	1.000	0.939	1.000	1.000
28	Ecoli-0-3-4-7_vs_5-6	0.964	0.966	0.998	0.983	0.996	1.000	1.000	0.991	1.000	0.991	0.987	0.985
29	Ecoli-0-3-4-6_vs_5	0.845	0.986	0.986	0.959	1.000	0.986	1.000	0.980	0.993	0.885	0.986	0.983
30	Vowel0	0.932	0.976	1.000	0.979	0.958	0.977	0.993	0.978	0.996	0.972	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
32	Ecoli-0-6-7_vs_5	1.000	1.000	0.991	0.981	1.000	1.000	1.000	1.000	1.000	1.000	0.994	0.981
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.761	0.941	0.742	0.828	0.938	0.812	0.884	0.935	0.952	0.886	0.742	0.817
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.918	0.962	0.896	0.954	0.964	0.934	0.924	0.960	0.957	0.958	0.899	0.936
35	Ecoli-0-1_vs_5	0.857	0.882	0.929	0.993	1.000	0.895	0.993	1.000	0.997	0.836	0.929	0.929
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.810	1.000	0.333	1.000	1.000	1.000	1.000	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.684	0.958	0.794	0.832	0.932	0.803	0.806	0.910	0.968	0.869	0.794	0.897
38	Ecoli-0-1-4-6_vs_5	0.868	0.981	1.000	0.962	0.995	0.962	1.000	0.971	1.000	0.844	1.000	1.000
39	Shuttle-c0-vs-c4	1.000	0.965	0.983	0.967	1.000	1.000	1.000	1.000	1.000	1.000	0.983	1.000
40	Page-blocks-1-3_vs_4	0.994	0.934	0.939	0.996	0.924	0.995	0.935	1.000	1.000	0.994	0.939	1.000
41	Glass4	0.625	0.750	1.000	0.936	0.974	0.949	0.071	0.981	0.801	0.955	1.000	1.000
42	Dermatology-6	1.000	0.979	1.000	1.000	1.000	1.000	0.993	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.734	0.891	0.500	1.000	0.984	0.969	0.844	0.938	0.969	0.891	0.500	1.000
44	Ecoli4	1.000	0.984	1.000	1.000	0.989	1.000	1.000	1.000	0.995	1.000	0.995	1.000
45	Zoo-3	0.671	0.750	1.000	0.658	0.763	1.000	0.132	0.921	0.697	1.000	1.000	0.750
46	Poker-9_vs_7	0.686	0.543	0.989	0.564	0.894	0.479	1.000	1.000	0.500	0.830	0.989	1.000
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
49	Yeast-2_vs_8	0.941	0.902	0.888	0.897	0.911	0.865	0.872	0.841	0.967	0.972	0.888	0.888
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.992	0.993	0.934	0.992	0.993	0.993	0.993	0.994	0.990	0.994	0.931	0.930
53	Ecoli-0-1-3-7_vs_2-6	1.000	0.991	1.000	0.982	0.982	0.946	0.964	1.000	1.000	0.938	1.000	0.991
54	Abalone-21_vs_8	0.896	0.895	0.695	0.975	0.888	0.941	0.939	0.849	0.793	0.857	0.695	0.798
55	Kddcup-land_vs_portsweep	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.500	0.325	0.495	0.504	0.406	0.690	0.253	0.432	0.521	0.531	0.495	1.000
57	Shuttle-2_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.996	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.999	1.000	1.000	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.998	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.912	0.922	0.930	0.939	0.945	0.945	0.800	0.958	0.949	0.950	0.932	**0.959**
	Avg. Ranks	3.567	3.967	4.133	4.933	5.467	4.983	4.317	5.967	**6.283**	5.600	4.117	5.300

Table A4
PR-AUC RESULTS COMPARISON.

Idx.	Dataset	DT	NB	k-NN	LR	RF	LinearSVM	RBF_SVM	Bagging	AdaBoost	XGB	mPEkNN	**EMass**
1	Glass1	0.795	0.472	0.779	0.372	0.632	0.432	0.242	0.656	0.537	0.807	0.850	0.921
2	Wisconsin	0.960	0.988	0.976	0.982	0.967	0.982	0.921	0.979	0.994	0.981	0.976	0.977
3	Pima	0.719	0.711	0.601	0.711	0.713	0.655	0.664	0.695	0.675	0.697	0.632	0.614
4	Ecoli-0_vs_1	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	Iris0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	Glass0	0.501	0.537	0.698	0.585	0.784	0.599	0.178	0.759	0.820	0.775	0.715	0.795
7	Vehicle2	0.879	0.708	0.917	0.964	0.904	0.966	0.659	0.914	0.988	0.995	0.910	0.980
8	Vehicle1	0.361	0.547	0.536	0.728	0.545	0.730	0.557	0.667	0.729	0.698	0.563	0.666
9	Glass-0-1-2-3_vs_4-5-6	0.913	0.898	0.966	0.958	0.937	0.979	0.201	0.980	1.000	0.948	0.974	1.000
10	Vehicle0	0.902	0.555	0.955	0.995	0.719	0.985	0.696	0.879	0.994	0.981	0.941	0.940
11	Ecoli1	0.777	0.573	0.710	0.811	0.676	0.832	0.753	0.783	0.731	0.785	0.761	0.816
12	New-thyroid1	1.000	1.000	1.000	1.000	1.000	1.000	0.930	1.000	1.000	1.000	1.000	1.000
13	Newthyroid2	0.944	1.000	1.000	1.000	1.000	1.000	0.901	1.000	1.000	1.000	1.000	1.000
14	Ecoli2	0.785	0.747	0.828	0.746	0.828	0.601	0.822	0.818	0.856	0.877	0.816	0.841
15	Segment0	0.983	0.954	0.977	1.000	0.954	1.000	0.896	0.971	0.994	0.988	0.977	0.987
16	Glass6	0.795	0.808	0.857	0.915	0.964	0.926	0.101	0.940	0.940	0.969	0.857	0.944
17	Yeast3	0.852	0.863	0.811	0.752	0.856	0.725	0.773	0.854	0.754	0.850	0.788	0.882
18	Ecoli3	0.851	0.727	0.756	0.714	0.772	0.675	0.881	0.803	0.741	0.911	0.829	0.807
19	Page-blocks0	0.818	0.621	0.860	0.814	0.782	0.692	0.617	0.772	0.901	0.891	0.864	0.887
20	Yeast-0-2-5-6_vs_3-7-8-9	0.442	0.418	0.676	0.459	0.761	0.474	0.584	0.671	0.413	0.686	0.603	0.606
21	Yeast-0-2-5-7-9_vs_3-6-8	0.603	0.656	0.877	0.632	0.889	0.654	0.911	0.863	0.760	0.810	0.851	0.872
22	Yeast-2_vs_4	0.883	0.869	0.864	0.894	0.947	0.886	0.919	0.959	0.983	0.919	0.834	0.886
23	Ecoli-0-6-7_vs_3-5	0.875	0.514	1.000	0.514	0.850	0.514	1.000	1.000	1.000	0.903	1.000	1.000
24	Ecoli-0-1_vs_2-3-5	0.438	1.000	1.000	0.708	1.000	0.792	1.000	1.000	1.000	0.597	1.000	1.000
25	Ecoli-0-2-3-4_vs_5	0.833	1.000	1.000	0.626	0.908	0.650	1.000	1.000	0.944	1.000	1.000	1.000
26	Ecoli-0-2-6-7_vs_3-5	1.000	0.455	1.000	0.579	1.000	0.544	1.000	1.000	0.796	0.944	1.000	1.000
27	Ecoli-0-4-6_vs_5	0.846	0.903	1.000	0.850	0.903	0.850	1.000	0.850	1.000	0.754	1.000	1.000
28	Ecoli-0-3-4-7_vs_5-6	0.775	0.675	0.983	0.835	0.963	1.000	1.000	0.938	1.000	0.938	0.920	0.843
29	Ecoli-0-3-4-6_vs_5	0.710	0.908	0.900	0.519	1.000	0.908	1.000	0.884	0.944	0.793	0.900	0.850
30	Vowel0	0.910	0.872	1.000	0.852	0.849	0.845	0.919	0.873	0.984	0.948	1.000	1.000
31	Glass-0-4_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.054	1.000	1.000	1.000	1.000	1.000
32	Ecoli-0-6-7_vs_5	1.000	1.000	0.946	0.835	1.000	1.000	1.000	1.000	1.000	1.000	0.944	0.835
33	Ecoli-0-1-4-7_vs_2-3-5-6	0.461	0.658	0.672	0.656	0.687	0.643	0.691	0.712	0.795	0.647	0.672	0.635
34	Led7digit-0-2-4-5-6-7-8-9_vs_1	0.825	0.789	0.819	0.791	0.791	0.764	0.762	0.830	0.781	0.834	0.807	0.795
35	Ecoli-0-1_vs_5	0.878	0.880	0.939	0.966	1.000	0.883	0.966	1.000	0.981	0.780	0.939	0.939
36	Glass-0-6_vs_5	1.000	1.000	1.000	1.000	0.100	1.000	0.033	1.000	1.000	1.000	1.000	1.000
37	Ecoli-0-1-4-7_vs_5-6	0.472	0.759	0.735	0.771	0.605	0.668	0.682	0.561	0.821	0.523	0.735	0.874
38	Ecoli-0-1-4-6_vs_5	0.821	0.579	1.000	0.462	0.944	0.462	1.000	0.842	1.000	0.782	1.000	1.000
39	Shuttle-c0-vs-c4	1.000	0.953	0.985	0.969	1.000	1.000	0.999	1.000	1.000	1.000	0.985	1.000
40	Page-blocks-1-3_vs_4	0.950	0.739	0.904	0.966	0.467	0.958	0.687	1.000	1.000	0.950	0.896	1.000
41	Glass4	0.660	0.155	1.000	0.399	0.579	0.442	0.051	0.767	0.775	0.804	1.000	1.000
42	Dermatology-6	1.000	0.700	1.000	1.000	1.000	1.000	0.792	1.000	1.000	1.000	1.000	1.000
43	Winequality-white-9_vs_4	0.515	0.587	0.529	1.000	0.792	0.708	0.149	0.633	0.417	0.466	0.529	1.000
44	Ecoli4	1.000	0.663	1.000	1.000	0.930	1.000	1.000	1.000	0.955	1.000	0.944	1.000
45	Zoo-3	0.399	0.774	1.000	0.551	0.570	1.000	0.053	0.663	0.557	1.000	1.000	0.774
46	Poker-9_vs_7	0.332	0.517	0.875	0.052	0.149	0.037	1.000	1.000	0.515	0.542	0.792	1.000
47	Shuttle-c2-vs-c4	1.000	1.000	1.000	1.000	1.000	1.000	0.019	1.000	1.000	1.000	1.000	1.000
48	Shuttle-6_vs_2-3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
49	Yeast-2_vs_8	0.809	0.739	0.762	0.845	0.717	0.769	0.711	0.625	0.521	0.778	0.762	0.680
50	Kddcup-guess_passwd_vs_satan	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
51	Abalone-3_vs_11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
52	Yeast5	0.803	0.733	0.648	0.779	0.887	0.777	0.759	0.841	0.635	0.852	0.626	0.733
53	Ecoli-0-1-3-7_vs_2-6	1.000	0.750	1.000	0.250	0.250	0.125	0.167	1.000	1.000	0.083	1.000	0.750
54	Abalone-21_vs_8	0.849	0.652	0.563	0.849	0.627	0.824	0.721	0.811	0.642	0.580	0.563	0.759
55	Kddcup-land_vs_portsweep	1.000	1.000	1.000	1.000	1.000	1.000	0.002	1.000	1.000	1.000	1.000	1.000
56	Poker-8-9_vs_6	0.215	0.012	0.008	0.018	0.013	0.030	0.010	0.016	0.017	0.024	0.008	1.000
57	Shuttle-2_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	0.359	1.000	1.000	1.000	1.000	1.000
58	Kddcup-buffer_overflow_vs_back	1.000	0.944	1.000	1.000	0.915	1.000	1.000	1.000	1.000	1.000	1.000	1.000
59	Kddcup-land_vs_satan	1.000	1.000	0.750	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	1.000
60	Kddcup-rootkit-imap_vs_back	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Avg. Values	0.819	0.776	0.878	0.795	0.819	0.800	0.697	0.880	0.865	0.852	0.875	**0.915**
	Avg. Ranks	4.250	3.600	5.067	4.517	5.000	4.567	4.017	5.767	5.917	5.817	4.933	**6.183**

Innovation of Mechanical Practice Model using Virtual Reality Technology and Pilot Butt Welding technique in Flat Position

Nguyen Van Huan

Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam

Correspondence: Nguyen Van Huan, nvhuan@ictu.edu.vn
Communication: received 29/11/2024, revised 10/02/2025, accepted 04/04/2025
Digital Object Identifier: 10.32913/mic-ict-research.v2025.n2.1358

Abstract: Metal welding is one of the indispensable jobs in daily practice. Nowadays, there are many different welding techniques and methods depending on the object/metal to be welded. This article focuses on studying the principle of butt welding technique of metal plates in the flat position using traditional tools, machinery and equipment, thereby proposing a solution to innovate the practice model of butt welding technique in the flat position with the traditional method on real equipment to the practice method on virtual reality technology. Accordingly, the practical method will design and build 3D models of traditional equipment, tools, and instruments such as soldering irons, current regulators, metal iron blanks, protective gear, etc. using 3D graphic design technologies such as Unity 3D, Maya 3D, 3Ds Max, etc. and build virtual simulation exercises of the metal welding process in the form of a visual virtual reality model using the OpenSG graphics library and C# programming languages. The results of the article will build a virtual reality practice model for butt welding metal plates in a flat position with instructions for automatically explaining each step of the metal welding process. At the same time, the article will also provide assessments and analysis of the quality of the practice exercises at vocational training institutions in Thai Nguyen province.

Keywords: *Butt welding, iron blanks, virtual reality, virtual simulation, 3D model*

I. Introduction

Education and training play a very important role for any country, especially in the current trend of educational streaming, which is professional education and vocational education. In vocational education, the orientation of transforming from training practical skills for people is an inevitable trend to innovate training methods, contributing to improving training quality. In the trend of digital transformation, the 4th industrial revolution has strongly impacted vocational education and training, in which, the main focus is on solutions, published research works, exploiting digital technology solutions into applications to change the way of training vocational skills.

However, to have a complete, objective and accurate view of the exploitation and application of digital transformation technologies, mainly the technologies of the 4th industrial revolution, to support the digital transformation activities of real models to virtual reality (VR) practice models. How is the reality? What is the effectiveness? It requires educational managers and scientists to analyze, evaluate and propose solutions to innovate vocational skills training models. This article is structured as follows: 1). General introduction; 2). Related research works; 3). Research methods; 4). Research results and 5). Conclusions and recommendations.

II. Literature Review

Digital transformation of vocational skills training models based on virtual reality technology is one of the solutions that has brought many effective results in recent years. Evidence shows that both in the world and in the country, it has received much research attention from scientists, experts and education and training managers. Initially, there have been studies and significant achievements in building and developing technological solutions in vocational training practice, specifically:

In the world, Farkhatdinov, I., & Ryu, J. H. (2009); George, A. (2021) have researched and tested the application of virtual reality technology to support mechanical automation practice lessons. The authors Cooper et al., (2019); Athri Nandan, (2020); Demitriadou et al. (2020) have researched the application of information technology, including the use of virtual reality technology to organize virtual classes and the results show the effectiveness and experience of learners. Scaravetti, D. & François, D., (2021) have focused on the research on the use of AR (Augmented

Reality) technology in mechanical engineering. This solution allows learners to learn and interact with lesson content conveniently. The research team applied AR to build solutions that allow the selection of AR devices and software, allowing for a digital chain integrated with the tools and digital files used by mechanical engineers according to Gavish et al. (2021) to practice more proficient skills. VR (Virtual Reality) is an educational technology with significant advances, the application of VR, AR is an important part of the development of professional skills adapted to the context of the technology that engineers must be trained. According to the group of authors (Tao Cheng, 2012; T*u*mler et al., 2021; Washington Quevedo, 2017), it is believed that the application of AR will enhance the interaction of the practice environment, allowing the creation of an effective technological platform for training in the field of mechanics. Applying AR to the design of virtual practice exercises will help learners increase their motivation and joy in learning, exploring, experiencing and visualizing complex phenomena. According to the groups of authors (Wang et al., 2018; Webel et al., 2011 & 2013), it is said that when using AR in practice, it will help learners concentrate more, increase the feeling of ”presence” and improve memory, especially creating advantages to increase learners' motivation and positive attitudes, improve understanding and learning performance, and increase learners' participation. Sabine Webel et al. (2012); Milgram, P. & Kishimo, F. (1994); Neumann, U.& Majoros, A. (1998), discussed the process of creating simulation scenarios for practical lessons, using the real/virtual continuum between tangible and VR interfaces combined with wearable devices to create practical lessons, 3D model representations to support the construction of components in practical lessons and detailed descriptions of the practical lesson process.

Thus, the world has received much attention from scientists and educational managers in vocational skills training activities. They have applied experimental digital technologies such as virtual reality technology to support the digital transformation of traditional vocational training models to virtual vocational training models.

Similar to the world, in Vietnam, vocational training activities in educational institutions have received much attention from many countries, scientists and managers are interested in developing new technology applications such as VR, AR in building simulation software for lessons, mechanical, electrical, welding practices... and have achieved many results. Vocational education and training also needs to change, change training methods and approaches from traditional practice lessons on machinery and equipment in rooms to practice on virtual software. This will partly overcome the current limitations of the lack of facilities and practice equipment. Vocational training facilities have outdated practice equipment, cannot keep up with the trend, and lack the exploitation and development of digital transformation technology applications, 4.0 technology in building virtual reality practice solutions. The project ”Virtual General Physics Laboratory”, by the Institute of Technical Physics - Hanoi University of Science and Technology, built virtual physics experiment software. With liquids, gases ... and equipment, tools used are completely designed, the process of scientific processing and image authenticity, the experimental operations are simulated ”almost” real. Creating a virtual reality practice environment on the computer according to the Department of Physics and Informatics, (2001). Mimbus Group, 2019], researched solutions to apply virtual reality technology to vocational education training and initially they have produced products such as: Smart classroom solutions, virtual reality simulation systems, supporting human resource training in the automotive mechanical and welding industries ... with applications designed to provide interactive, hands-on learning experiences. Mechanical engineering simulation system: allows learners to explore mechanical components with 3D simulation objects to develop the fundamentals needed in manufacturing, logistics and maintenance processes, Pneumatic system and equipment simulation system: uses virtual reality models, animations to help learners understand the detailed structure and overall structure of the pneumatic system. The welding practice simulation system runs on the zSpace platform, allowing learners to practice arc welding, with exercises from easy to difficult according to European standards. The most effective simulation welding machine for learning and practicing welding motion skills, the solution allows depending on the selected practice lesson, the software is capable of simulating welding processes. Author groups Nguyen Duong, 2021; V. H. Nguyen and D. T. Vu, (2006) also studied virtual reality technology and initially tested the installation of virtual reality simulation technology applications for a number of occupations, including mechanical engineering. However, the results are only at the research and testing stage, especially the documents used for teaching in technical universities, which are the University of Information Technology and Communications - Thai Nguyen University.

For Thai Nguyen province, where there are many vocational training institutions such as: Thai Nguyen College of Economics and Technology; Viet Duc Industrial College; College of Technology and Commerce; Thai Nguyen Industrial College; Thai Nguyen Vocational College... Through actual surveys, it shows that most training institutions still mainly use traditional equipment and practice machines and some schools still lack practice equipment due to high

purchase costs. Up to now, in Thai Nguyen, there has not been any institution that has applied VR and AR technology to support vocational skills practice for learners. These are shortcomings and limitations that require effective investment solutions and changes in vocational skills training models for learners.

Thus, with the analysis both in the world and in the country, it is shown that many educational institutions still do not fully meet the current practical requirements, due to many different reasons such as insufficient finance, not enough students to be able to invest or equip more. Therefore, finding new solutions that are economical and suitable for the reality at each vocational training institution is very necessary.

Within the scope of this paper, only one practical lesson has been selected to pilot the application of virtual reality (VR) technology in building a virtual reality practice lesson with metal welding techniques in the flat position to enhance the practice and experience of learners with welding techniques. Flat position butt welding (symbol - 1G) is widely applied in practice with the advantage of high welding productivity. Therefore, if conditions permit, we should switch to the flat position for welding. Having the skill to butt weld in the flat position will be the initial step in developing the skill. Flat position butt welding is often used in steel structures, where two parts are joined together by welding (Gas Welding Lecture, 2015; M. H. Nguyen, 2012; N. D. Pham & Q. H. Nguyen, 2007). The results of this pilot will be researched, replicated and tested for virtual reality practice with other techniques such as welding, electricity or mechanics ...

III. METHOD OF BUTT WELDING METAL JOINTS IN FLAT POSITION

1. Method Introduction

Flat position butt welding is a simple welding method widely used in electrical engineering, radio, welding of metal cutting tools, thermal tools, household appliances. Flat position butt welding is the most popular welding method in all welding positions. This is considered a basic weld that any welder must learn.

For flat position butt welding technique, note the following: Depending on the thickness of the welding operation, different welding rods and welding current settings will be selected. You can read more detailed instruction manuals, for example, if welding 3.2 mm rods, the current should be 120 - 140 A.

During the welding process, it is important to keep the arc length stable, oscillating evenly, and not too fast at the edge positions.

The welder can choose the straight or sawtooth welding method. If welding in a straight line, the arc is concentrated in the middle of the weld, so the penetration will be good. If welding in a sawtooth oscillation, attention must be paid to the welding speed to ensure the width of the weld and there must be a stop at the beginning and end points to achieve edge penetration.

For welding materials > $6mm$ thick or more, to ensure the melting depth, it is necessary to bevel the edges and weld multiple layers. When welding multiple layers, choose a welding rod with a small diameter to weld the first lining layer, move in a straight line, and for the second layer, choose a larger welding rod diameter, move in a straight line or sawtooth, use a short arc to weld. Note that each layer of weld should not be too thick.

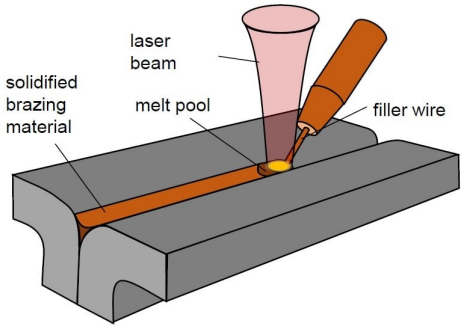


Figure 1. Metal welding model by butt welding in flat position.

For flat position butt welding is often used in steel structures, where two parts are joined together by welding. To ensure the durability and stability of the weld, you need to pay attention to the following factors:

1. **Welding type**: Spot welding, continuous welding or U-shaped welding can be used depending on the technical requirements and load of the structure.
2. **Weld size**: Weld size needs to be calculated to have sufficient strength according to design standards.
3. **Surface preparation**: The surface of the parts needs to be cleaned to ensure good weld adhesion.
4. **Welding procedure**: Standard welding procedure must be followed, including temperature adjustment, welding speed and welding time.
5. **Quality check**: After welding is complete, the weld should be inspected to ensure there are no defects such as cracks, porosity or missing weld material.

For flat position metal butt welding technique, to ensure weld quality, welding engineers need to pay attention to choosing:

+ **Welding rod angle**: The welder needs to maintain the correct welding rod angle and stable arc length (Lhq=2 ~ 4 mm) throughout the welding process; Correctly perform the welding seam connection operation; When the welding seam is finished, the weld pool must be filled.

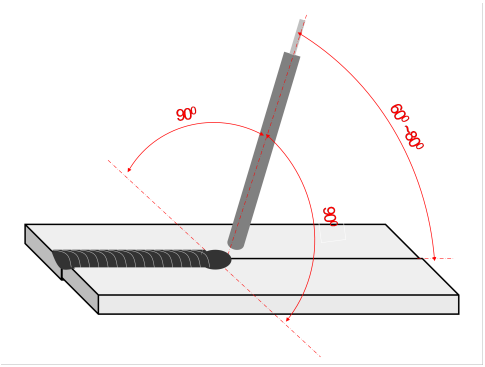


Figure 2. Illustration of welding rod angle.

+ **Transverse oscillation of the welding rod**: To ensure the weld penetration, when welding the welding rod can oscillate in a straight line or a sawtooth oscillation. If it follows a straight line, the arc is concentrated in the middle of the weld, so the penetration in this case is better. When oscillating in a sawtooth shape, the welding speed must be appropriate (to ensure the weld width) and there must be a stop on both sides to achieve the weld edge penetration.

2. Butt welding technique of metal joints in flat position

To weld two metal plates/iron pieces into one, we need to follow the following sequential steps:

- Step 1- Preparation: Welding table, iron piece, welding head, ground wire, welding machine, welding rod, protective equipment, other items. Check and install equipment. Make sure the equipment is working well and safely and is checked periodically.
- Step 2- Determine the weld line: This can be done using a template or marking directly on the metal surface.
- Step 3- Connect and adjust the welding machine: Connect and adjust the current to suit the welding workpiece as well as the welding rod with the equipment, ensuring that the equipment wires are connected together.
- Step 4- Fix the welding blank: welding to avoid the welding blank from shifting, also to ensure the best welding line, correct technique and high aesthetics.
- Step 5- Welding: tilt the welding rod tip at a 65 degree angle to the welding workpiece, dot each point along the edge of the 2 pieces of workpiece at the welding position, each position overlaps about 50%.
- Step 6- Welding slag removal process: After welding is complete, turn off the equipment to ensure safety and remove the welding slag at the welding site, exposing the weld.
- Step 7- Check the results: Check the weld to ensure that it meets the requirements of the exercise and has no unwanted marks.

Some notes on butt welding instructions in flat position:

How to choose solder scales: When choosing solder scales, you need to base on the technical conditions of the weld as well as the working environment of the object to be able to choose the appropriate solder scales for the requirements of the weld.

Soldering mode: In soldering mode, the most important factors are the welding temperature, welding time, melting temperature of the solder and solder scales, and finally the heating rate.

- The welding temperature is a definite quantity and the welding temperature will be greater than the temperature of the solder.
- The heating time will depend on the factors such as the size of the solder, the gap between the solder, the metal composition of the solder and the solder.
- The heating speed will depend on the size of the solder, the thermal conductivity of the solder and the requirements of the welding technique.

Heating method: To have a beautiful weld, it is necessary to ensure 3 factors: good weld structure, suitable weld scale and good heating method. Therefore, the heating method is extremely important. If the weld structure is good, the weld scale is suitable but the heating is not good, the quality of the weld will also be poor.

For the best quality weld, when heating, it is necessary to heat evenly on all sides of the welding machine, the weld and the weld scale.

Weld structure: The important factor determining the strength of the weld is the weld cross-section and the adjustment of the connections together.

IV. Results

1. Collect image data about tools, equipment, and facilities

The lesson on butt welding of metal joints in the flat position usually uses the tools, apparatus or equipment as shown Figure 4.

In addition to the above tools, instruments, and equipment,

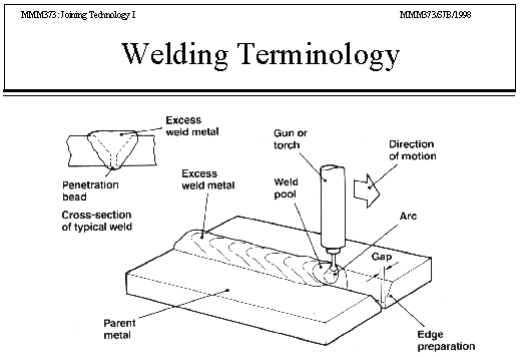


Figure 3. Weld structure.

Figure 4. Some tools and equipment commonly used in flat position butt welding.

many other equipment are also used to practice the lesson of welding metal joints in the same position. Figure **??** illustrates some of the above equipment and instruments, which the research team collected images and models using scanners, 3D scans, and cameras to take pictures.

2. Design 3D models of tools, instruments, and components used in virtual practice

Based on the images of tools, equipment, and machinery components used in the practice of butt welding metal joints in flat positions, perform the following main steps:

Figure 5 illustrates the process of creating 3D models of tools, equipment, and facilities:

Based on the images of tools, equipment, and machinery components used in the practice of butt welding metal joints in flat positions, perform the following main steps:

Step 1: Map the actual photos into graphics software such as Photoshop, 3Ds Max, Maya 3D... to process noise, fine-tune each component to ensure the actual size of the tool.

Step 2: Build 3D models of tools such as tables, iron blanks, etc.

Step 3: Perform processing, mesh optimization and material coating (texture) on the surface of each tool model.

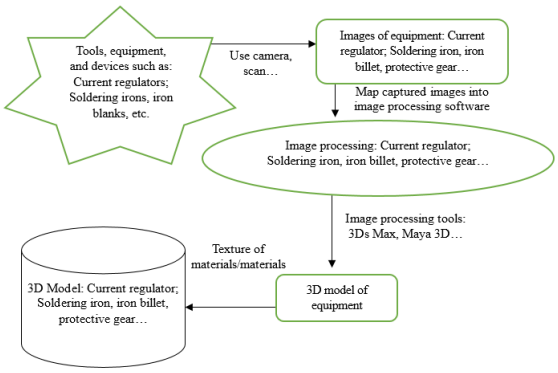


Figure 5. Virtualization process, creating interactive 3D models.

Step 4: Evaluate the 3D model results with images of each actual tool to see if the quality is guaranteed.

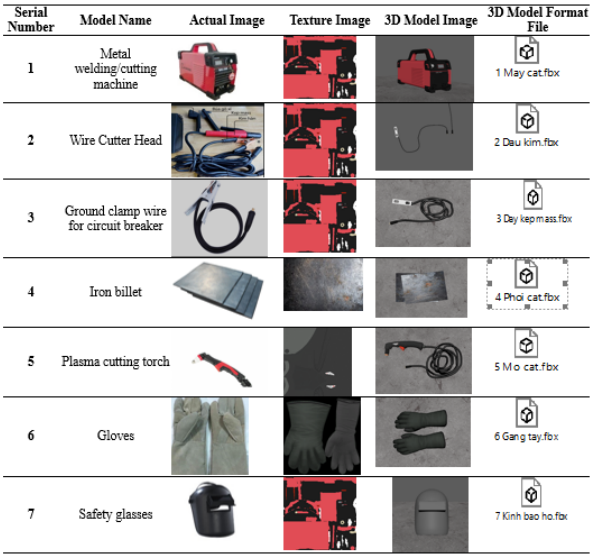


Figure 6. Some 3D models of tools built for use in the practice of butt welding metal joints in flat position.

With 3D models of each tool, apparatus, and component used in the metal butt welding practice in a flat position (Figure 6), the group of authors will rely on the scenario and teaching plan of the practice to design and build a virtual practice using Unity 3D technology and C# programming.

3. Solution to convert number of practice exercises on metal butt welding in flat position

With 3D models of tools, equipment, and equipment used in the practice of welding metal using gas welding techniques, the group of authors proceeded to build steps to convert the practice from traditional form to virtual reality practice using virtual reality programming technology. With the technique of welding metal blanks in the flat position,

it is modeled by the algorithm flowchart below Figure 7. Below are some simulation results illustrating in detail in

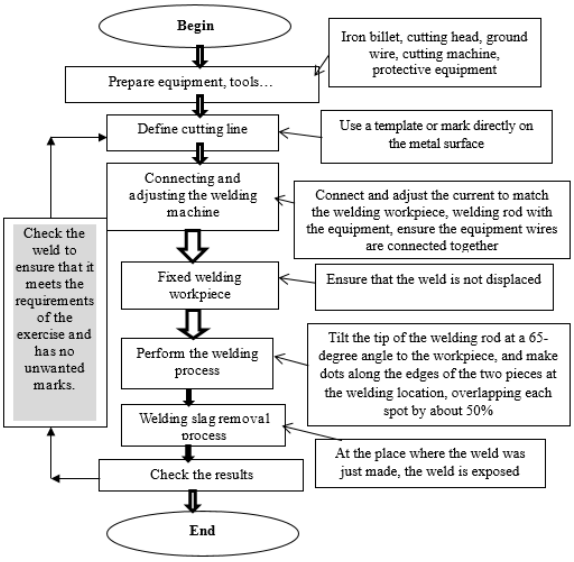


Figure 7. Algorithm flowchart simulating the metal butt welding process in flat position.

Figure 7 the progress of performing the metal butt welding practice in position using virtual reality technology. First step: Prepare the necessary 3D tools and equipment used in the exercise (Figure 8); determine the welding line and connect the current adjustment . . .

Figure 8. List of 3D tools and instruments.

3D models of tools and equipment: 3D models of tools, equipment and components including Electric arc metal welding machine, Wire welding pliers, Iron workpiece, Plasma welding torch, gloves, welding glasses... have been built from Figure 6.

Next steps: Perform metal butt welding in a flat position along the predetermined weld line.

Based on the steps of the metal butt welding method in the flat position and based on the algorithmic flowchart simulating the metal butt welding process in the flat position in Figure 7, the research group proceeded to build a virtual practice using virtual reality technology and other technologies including:

1) The group of software tools for creating 3D models and tools includes 3Ds max used for creating shapes, Zbrush used for creating shapes and convexities ..., Subtance Painter used for creating materials and Maya used for creating movement and animation.
2) The programming technology group builds virtual, interactive practice lessons including C#, Unity 3D Engine, VRML... used to program the control of initializing 3D models into the virtual reality environment and program the control of camera settings, interacting with 3D tools, and virtual practice process steps.

Below is a diagram detailing the process from preparation to the final step to achieve welding results:

The above process diagram describes each specific step

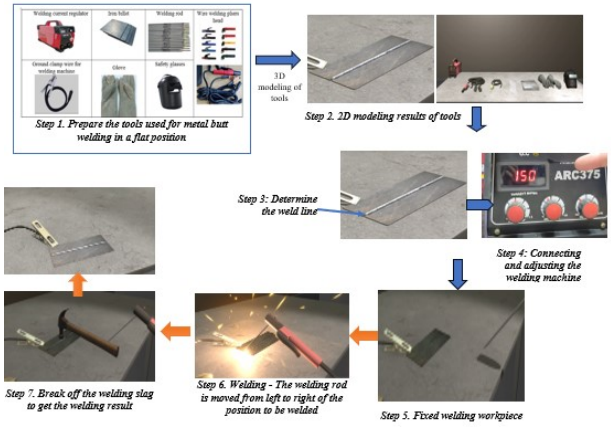


Figure 9. Diagram detailing the steps of the metal butt welding process in the flat position from the start to the result of the butt welded metal plate.

in detail with illustrations of the results of the metal butt welding process in the flat position. In particular, the process ensures that the main steps of the metal butt welding method are followed correctly.

Below is an illustration of the resulting sheet metal after welding:

4. Evaluation of virtual simulation program for metal butt welding practice in flat position

Part 4 presented the construction of a virtual simulation program for the practice of metal butt welding in a flat position, which was built on the basis of studying the content of traditional practice lessons, from which multimedia tools such as cameras, scanners, etc. were used to collect tools, instruments, and related components used in

Figure 10. Image of the butt-welded metal plate in the flat position after the slag has been removed.

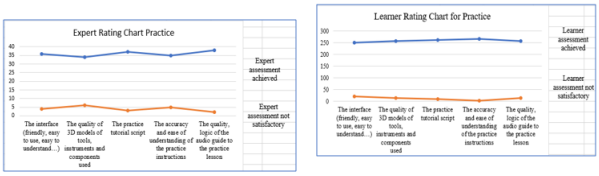
the practice lesson, and to build a 3D model of each tool and component. On that basis, the design, arrangement, and assembly of the virtual reality practice of metal butt welding in a flat position were carried out.

In order to evaluate and test the virtual reality practice on metal butt welding in a flat position, the group of authors has built a group of 5 criteria on friendliness, ease of use, model quality, practice instruction script, etc. The group of authors has designed a survey form to distribute and investigate scientists, education experts and related learners. Through organizing a scientific workshop, presenting in detail the virtual reality practice software on metal butt welding in a flat position. Then, the survey form is distributed. At the workshop, the group of authors invited about 40 experts, scientists, teachers with specialized knowledge related to welding, electrical, mechanical engineers, etc. and 270 learners from 5 colleges and vocational schools in Thai Nguyen province that provide training in electrical, mechanical, welding, etc.

The scientific workshop focused on presenting two main topics: an overview of the content of the traditional practical lesson teaching script and running a program introducing the main functions guiding the virtual vocational training skills practice of metal butt welding in a flat position on a simulation software environment. The results received were evaluated by 40 experts, scientists, teachers and 270 students from 5 colleges and vocational schools in Thai Nguyen province that provide training in electricity, mechanics, welding, etc. The evaluation results and comments are illustrated in Figure 11 and Figure 12. Thus, the

Survey criteria	Expert assessment achieved (%)	Expert assessment not satisfactory	Learner assessment achieved (%)	Learner assessment not satisfactory
The interface (friendly, easy to use, easy to understand…)	36 (90%)	4	250 (93%)	20
The quality of 3D models of tools, instruments and components used	34 (85%)	6	257 (95%)	13
The practice tutorial script	37 (93%)	3	262 (97%)	8
The accuracy and ease of understanding of the practice instructions	35 (88%)	5	267 (99%)	3
The quality, logic of the audio guide to the practice lesson	38 (95%)	2	256 (95%)	14

Figure 11. Summary of survey results.



a) Expert assessment survey b) Student assessment survey

Figure 12. Evaluation chart of virtual reality practice simulation system.

chart in Figure 12 shows that the virtual reality simulation system for metal butt welding in a flat position ensures the evaluation criteria of quality, interface, teaching scenario... ensuring quality and is highly evaluated. Figure 11 also shows the results of the evaluation survey of experts and learners with a high rate of evaluation of quality such as:

- Regarding the interface (friendly, easy to use, easy to understand...): Up to 90% of experts, scientists and teachers rated it as satisfactory; up to 93% of learners rated it as satisfactory, the remaining 7% of learners suggested that it should be adjusted to be more friendly.

- Regarding the quality of 3D models of tools, instruments and components used in the practice: Up to 85% of experts, scientists and teachers assessed the 3D models as satisfactory; up to 95% of learners assessed the 3D models as satisfactory, the remaining 5% of learners should suggest further revisions.

- Regarding the practice instruction script: Up to 93% of experts, scientists and teachers assessed the instruction script as satisfactory; up to 97% of learners assessed the instruction script as satisfactory, the remaining 3% of learners assessed it as unsatisfactory.

- Regarding the accuracy and ease of understanding of the practice instructions: Up to 88% of experts, scientists and teachers assessed it as satisfactory; up to 99% of learners assessed it as satisfactory, the remaining 1% of learners should suggest further revisions.

- Regarding the quality and logic of the audio guide to the practice lesson: Up to 95% of experts, scientists and teachers assessed that it met the requirements; up to 95% of learners assessed that it met the requirements, while the remaining 5% of learners assessed that it did not meet the requirements.

V. Conclusion

Vocational training is one of the important tasks for any country and for our country it is even more important and education and training in general has become a top national policy. Over the years, our country's education and training have changed a lot, contributing a large amount of human resources and labor to the country. And in particular,

education has had changes in development and application of achievements of science and technology to promote development, approach innovation and creativity. Besides the work that has been done in education in Vietnam in general and vocational education in particular, we need to have many more changes. Approaching education in a vocational direction, meeting social needs and needing to change the methods and ways of training theory combined with vocational practice based on technology and reality in enterprises.

This article focuses on researching and proposing solutions to apply digital transformation technology and virtual reality to build a virtual practice model on a technology platform using graphic design software such as Unity 3D, 3Ds Max, Maya 3D and control programming languages such as C#... The article has chosen to pilot the construction of a virtual reality practice lesson for the metal butt welding profession in a flat position. The results of the virtual reality practice lesson have been evaluated by 40 experts, scientists, and teachers currently teaching at vocational education institutions and tested for 5 classes with 270 learners asked for their opinions. The evaluation results have been summarized, analyzed and commented on in section IV.4, showing that the virtual simulation system of the metal butt welding practice in the flat position has met the requirements and ensured quality. Through the virtual reality practice system, learners can grasp knowledge and approach the application of the practice faster, mastering the practical skills of the lesson better.

ACKNOWLEDGEMENT

This article is a product of the 2024 ministerial-level project: "Research on developing a practical model of teaching smart mechanical engineering based on the application of virtual reality technology for students in mountainous and ethnic minority areas"; code: B2024-TNA-05 by the project leader, Dr. Vu Duc Thai.

REFERENCES

[1] A. Nandan, "Best software courses for mechanical engineering students," Skyfi Education Labs, 2020. [Online]. Available: https://www.skyfilabs.com/blog/best-software-courses-for-mechanical-engineering-students

[2] G. Cooper, H. Park, Z. Nasr, L.-P. Thong, and R. Johnson, "Using virtual reality in the classroom: Preservice teachers' perceptions of its use as a teaching and learning tool," *Educational Media International*, vol. 56, pp. 1–13, 2019.

[3] E. Demitriadou, K. E. Stavroulia, and A. Lanitis, "Comparative evaluation of virtual and augmented reality for teaching mathematics in primary education," *Education and Information Technologies*, vol. 25, pp. 381–401, 2020.

[4] Department of Physics and Informatics, "Application of IT in teaching," Center for High Performance Computing, ITIMS Informatics Group, Hanoi, 2001.

[5] I. Farkhatdinov and J. H. Ryu, "Development of an educational system for automotive engineering based on augmented reality," in *Proc. Int. Conf. Engineering Education and Research*, 2009.

[6] A. George, "Lockheed is using these augmented reality glasses to build fighter jets," *Popular Mechanics*, 2021. [Online]. Available: https://www.popularmechanics.com. [Accessed: Aug. 1, 2021].

[7] "Gas welding lecture – Lesson 2.2: Butt welding using gas welding method in flat welding position," 2015. [Online]. Available: https://tailieu.vn/doc/bai-giang-han-khi-bai-2-2-han-giap-moi-bang-phuong-phap-han-khi-o-vi-tri-han-bang-2561176.html

[8] N. Gavish *et al.*, "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks," *Interactive Learning Environments*, vol. 23, pp. 778–798, 2015.

[9] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Information Systems*, vol. 77, pp. 1321–1329, 1994.

[10] M. H. Nguyen, "Lecture on manufacturing technology," College of Economics and Technology, Thai Nguyen University, 2012.

[11] U. Neumann and A. Majoros, "Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance," in *Proc. IEEE Virtual Reality Annual Int. Symp.*, Atlanta, GA, USA, pp. 4–11, Mar. 1998.

[12] D. Nguyen, "Application of VR/AR/MR technologies in modern industries," 2021. [Online]. Available: https://onetech.vn/blog/ung-dung-cong-nghe-vr-ar-mr-cho-cac-nganh-nghe-13548

[13] M. Nebeling *et al.*, "XRDirector: A role-based collaborative immersive authoring system," in *Proc. CHI Conf. Human Factors in Computing Systems*, Honolulu, HI, USA, Apr. 2020.

[14] N. D. Pham and Q. H. Nguyen, *Textbook on Machine Manufacturing Technology*. Hanoi, Vietnam: Hanoi Publishing House, 2007.

[15] D. Scaravetti and D. Doroszewski, "Augmented reality experiments in higher education for complex system appropriation in mechanical design," in *Proc. CIRP Design Conf.*, Póvoa do Varzim, Portugal, May 2019.

[16] S. Webel *et al.*, "An augmented reality training platform for assembly and maintenance skills," Fraunhofer IGD, Germany, 2012. [Online]. Available: https://mimbus.vn. [Accessed: Jun. 25, 2019].

[17] T. Cheng and J. Teizer, "Real-time resource location data collection and visualization technology for construction safety and activity monitoring," *Automation in Construction*, 2012.

[18] J. Tumler *et al.*, "Mobile augmented reality in industrial applications: Approaches for solving user-related issues," in *Proc. IEEE/ACM ISMAR*, Cambridge, UK, pp. 87–90, Sep. 2008.

[19] V. H. Nguyen and D. T. Vu, *Real-world simulation programming techniques based on Morfit 3D*. Hanoi, Vietnam: Science and Technology Publishing House, 2006.

[20] W. Quevedo, "Teaching–learning process through VR applied to automotive engineering," in *Lecture Notes in Computer Science*, 2017, doi: 10.1007/978-3-319-60922-5_14.

[21] M. Wang *et al.*, "Augmented reality in education and training: Pedagogical approaches and illustrative case studies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, pp. 1391–1402, 2018.

[22] S. Webel *et al.*, "Augmented reality training for assembly and maintenance skills," *BIO Web of Conferences*, vol. 1, p. 97, 2011.

[23] S. Webel *et al.*, "An augmented reality training platform for assembly and maintenance skills," *Robotics and Autonomous Systems*, vol. 61, pp. 398–403, 2013.

Nguyen Van Huan Nguyen Van Huan, received his PhD in mathematical assurance for computers in 2013, received the title of Associate Professor in 2017. Research fields are simulation technology, virtual reality, image processing and digital transformation into economics, culture and society. Email: nvhuan@ictu.edu.vn

Enhancing Recommender Systems: A New Approach Using Collaborative Filtering with Bayesian Optimization and Gaussian Processes

Tuan-Anh Nguyen
Ho Chi Minh City University of Foreign Languages - Information Technology, Ho Chi Minh City, Vietnam
Correspondence: Tuan-Anh Nguyen. Email: anhnt1@huflit.edu.vn
Communication: received 19/11/2024, revised 23/02/2025, accepted 25/04/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1348

***Abstract*: Recommendation systems frequently make use of collaborative filtering (CF). CF derives its strength from the fact that in order to profile its users, it does not require a lot of knowledge about them, but it depends on earlier users' ratings in which they were choosing products to recommend. Although there has been progress in modeling users as well as items, calibrating CF algorithms' hyperparameters will continue to be a difficult task. This paper has come up with a new way of doing this by the use of Bayesian optimization with Gaussian processes throughout hyperparameter tuning. The method in question automatically works on hyperparameters to make two fundamental and straightforward CF algorithms have solid results against three famous datasets (Netflix Prize, MovieLens 1M, MovieLens 10M), and two other datasets (Douban and Jester dataset 2). This solution not only exhibits solid performance in controlled experiments but also provides a simplified approach for practitioners, minimizing time-intensive manual adjustments while ensuring low overhead, therefore necessitating relatively low computational and developmental resources. This results in expedited deployment and simplified integration into practical systems, hence enhancing the dependability and scalability of recommendation engines.**

Keywords: *Bayesian optimization, collaborative filtering, Gaussian processes, recommender system.*

I. Introduction

Today consumers face a broad selection of online platforms offering e-commerce services and content. This multitude of options makes it challenging to match products with individual consumer tastes. All corporations operating in diverse industries such as Amazon, Google, and Netflix have found that personalized recommendation systems are of absolute importance to them. By employing these technologies, organizations can personalize their products to suit different consumer tastes therefore increasing their levels of satisfaction and loyalty [1–3].

Collaborative filtering (CF) is now the major method used for creating recommendation systems. In contrast to systems that require detailed user profiles, CF is based strictly on past user activity such as ratings and purchases. Therefore it does not need any domain-specific knowledge or collection of vast amounts of data to give meaningful recommendations [4]. Furthermore, through user behavior, CF is able to unearth hidden trends or customer preferences that are not captured by traditional approaches. Such efficiency has been demonstrated in Amazon's and Netflix's commercial use [5–7].

Despite the progress made in modeling techniques, hyperparameter tuning in CF systems has largely remained a human activity. Often enough, researchers base their hyperparameter choice on experience and experimentally established criteria [8]. This is usually time-consuming and not always the best way; for instance, tuning hyperparameters was crucial to success in the Netflix Prize competition [9, 10].

Several effective applications for hyperparameter optimization have shown favorable outcomes using Bayesian optimization with Gaussian priors [11–13]. Nonetheless, recent advancements in intricate deep learning architectures [14, 15] have highlighted the expanding potential of neural networks for CF, although with heightened computing demands and considerable tuning complexity. Conversely, our methodology emphasizes hyperparameter optimization in two principal collaborative filtering algorithms using Bayesian optimization utilizing Gaussian processes. We want to automate the tuning process to reconcile manual approaches with resource-intensive deep learning models, minimizing human interaction while possibly improving performance on commonly used datasets. This study illustrates the effectiveness of our methodology, yielding competitive outcomes on the Netflix Prize,

MovieLens 1M, MovieLens 10M, Jester and Douban datasets (refer to our source code in the GitHub repository https://github.com/anhnt1/netflix).

The paper is organized as follows. In the next section, this paper shall review the extant literature focusing particularly on the manual tuning of hyperparameters and how previous research dealt with modeling users and items along with their interactions. The following part will then provide a rationale for employing Bayesian optimizations through Gaussian processes. Section 4 explains our methodology; while Section 5 presents results from running both CF algorithms on the Netflix Prize, MovieLens 1M, Movialens 10M, Jester and Douban datasets. Finally, we summarize the findings.

II. Literature Review

Notable has been the recommendation systems research with a major focus on modeling people, items and their interactions. Also, in this area, a significant study has been carried out into a number of ways to improve its efficiency leading to a large number of research works in this direction. CF, on the other hand, has been an overarching technique in such studies. An early paper by Koren et al. [9] among others was based on matrix factorization methods applied for user-item interaction. These models represent the interaction matrix between users and items as latent feature vectors which are then employed to predict future preferences of users towards different items. To build more complicated models, several research works have introduced hybrid techniques that combine content-based filtering with CF for exploiting both user behavior and item properties [16, 17]. In the CF industry, these are standard matrix factorization techniques, without implying they are easy – such decompositions have high computational complexity; yet they are scalable, for instance, common Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF). This necessitated the Truncated–ULV decomposition technique (T–ULVD) as an efficient computational alternative for scaling [18]. Data sparsity is overcome through knowledge-enriched CF systems reported by [19] – in order to represent interactions between users and items, they model user attributes as well as item attributes obtained by knowledge graphs (KGs). The methodology followed in [20] incorporates trust links between users, thus addressing cold-start issues as well as data sparseness in CF models. Deep learning methods have recently been used to improve the performance of CF systems since they allow for their use with data sparsity. Advanced complex user-item interactions and latent attributes can now be learned by employing methods like Collaborative Deep Forest Learning (CDFL) and Multi-model deep learning. These incorporate deep neural networks for improved handling of data sparseness and cold-start problems in addition to enhancing recommendation accuracy by combining traditional CF methods with deep neural networks [14, 15]. Attention mechanisms and recurrent neural networks enable the capture of relevant information and long-range dependencies, thereby enriching feature representation hierarchies with more contextual information [21]. To estimate missing values in a user-item interaction matrix for better prediction of users' preferences, some models use Bayesian statistics for matrix completion or imputation via matrix factorization [8, 22, 23]. Consequently, just like any other study that focuses on the estimation of parameters only for CF models, their main goal is still the same.

Even with the substantial progress made in developing advanced CF models over the last two decades, hyperparameter optimization remains crucial for their improvement. In general, traditional approaches usually entail manual tuning that often takes up time and may lead to suboptimal outcomes due to the high-dimensional search space as well as complex interactions between these parameters. This complicates the process of tuning for model correctness, generalization, and computational efficiency [8, 24]. They could be learning rates, regularization coefficients, embedding dimensions, or neural-networks-based considerations in architectures [25, 26]. The manual approach is particularly challenging for newcomers in the area who lack adequate expertise.

To address hyperparameter tuning difficulties, automated approaches such as Bayesian optimization provide an alternative that enhances model accuracy while reducing modeling time by doing all the hard work for one. These methods employ optimization and machine learning in exploring and evaluating hyperparameter space and are often more effective than manual nursing [27, 28]. Various practical examples demonstrated the effectiveness of automated hyperparameter tuning techniques [11–13] and this can be a fruitful area of research in recommender systems. This research work is more certain about the practical significance of recommendation systems and CF models in developing automatic parameter-tuning techniques with our access to actual performance results. This paper sets out to explore an alternative way of tuning hyperparameters that is simple and applicable in CF systems with the potential for good performance. This might reduce the dependence on human effort as well as improve the algorithm's performance across diverse scenarios. Making it easier for more practitioners to deploy and customize successful recommender systems could potentially drive innovation within this realm by simplifying model optimization efforts for all

involved parties.

III. Bayesian Optimization with Gaussian Processes

1. Bayesian Optimization for Hyperparameter Tuning in CF

Hyperparameter tuning in CF is a good match for Bayesian optimization using Gaussian processes (GPs) because it effectively manages those expensive evaluations at the first instance. In hyperparameter tuning especially for CF models, Bayesian optimization focuses on maximizing (or minimizing) functions that are costly to evaluate. For example, projecting the behavior of the function using less number of evaluations facilitates an effective exploration within the parameter space by using GPs [29]. Combining Bayesian optimization with GPs may instead offer a more realistic description of the underlying function which could result in increased expressiveness of surrogate functions approximating complex models like those of CFs [29].

Secondly, it gives an ability to measure uncertainty. Indeed, GPs themselves can indicate how uncertain they are about their own projection hence serving as the basis for choices on the next sample's future location. This is valuable when the main goal of tuning hyperparameters is finding the most suitable values with few evaluations [30]. Because acquisition functions based on uncertainty increase convergence rates and accuracy in surrogate models which lowers risks associated with overfitting and in turn enhances CF task performances [30].

Thirdly, it enables adaptive exploration and exploitation in a single breath. Natural exploration-exploitation tradeoff within Bayesian optimization can enable one to move effectively over the hyperparameter space. The use of acquisition functions in directing the search process helps to achieve this balance [29]. Common random numbers can enhance the efficiency of Bayesian optimization by reducing variance and making the optimization process more robust [31].

In conclusion, using Gaussian priors for Bayesian optimization provides an appropriate approach for tuning hyperparameters in CF models because it manages costly evaluations effectively, gauges uncertainty, and balances exploration versus exploitation leading to precise hyperparameter adaptation as well as improved CF performance metrics.

For subsequent paragraphs, this paper plans to provide an overview of the Gaussian process which is used as a prior in Bayesian optimization. This will assist one grasp how this method is implemented and certain techniques behind its capabilities.

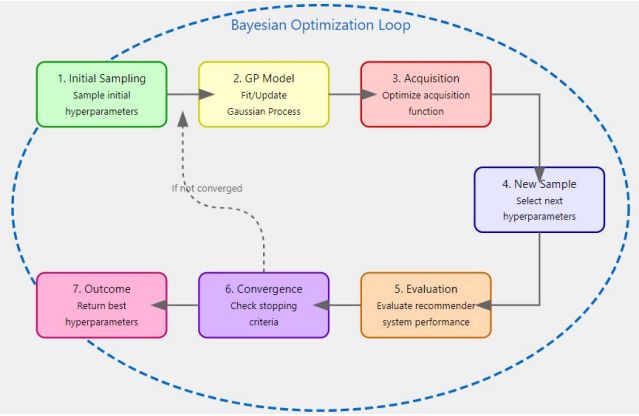


Figure 1. Bayesian Optimization Process for CFs.

2. High-Level Representation of Bayesian Optimization Process for Recommender Systems

Figure 1 illustrates the iterative characteristic of Bayesian optimization via a loop covering the whole process. Each iteration adjusts according to all prior assessments, gradually refining the best hyperparameters. When convergence has not been attained, the dashed arrow coming from the convergence check shows the iterative feature of the process.

1) Initial sampling: First, the method samples starting hyperparameters from the hyperparameter space. In this step, a limited number of hyperparameter combinations is either randomly or methodically chosen from the comprehensive search space. This first sample offers an initial insight into the model's potential responses to various parameter combinations.
2) Gaussian Process model: Second, Bayesian statistics is used in this step. It uses all available data (including the performance results from the initial samples) to fit a GP model, and new observations update this model in the next iterations. This step updates our beliefs to form a posterior distribution and uncertainty estimates. This GP serves as a proxy for the underlying (unknown) performance function, enabling us to estimate both the anticipated performance value (the mean) and the corresponding uncertainty at each location inside the hyperparameter space.
3) Acquisition function: Third, the present GP model is used to build an acquisition function that directs the selection of the subsequent hyperparameters for evaluation. This function addresses two objectives: (i) the exploitation of regions anticipated to provide high performance, and (ii) the investigation of areas characterized by significant model uncertainty. Typical acquisition functions include Probability of

Improvement (PI), Expected Improvement (EI), and Lower Confidence Bound (LCB). Readers should see the section III.3 for more details of these three acquisition functions.

4) New sample: Fourth, a new hyperparameter configuration is chosen from the search space based on the acquisition function's suggestions. This decision generally seeks to optimize possible improvement while also considering areas with significant uncertainty.
5) Evaluation: Fifth, the selected hyperparameter configuration is then assessed on the recommender system. This assessment entails training or refining the model using a dataset (e.g., Netflix Prize, MovieLens,...) and evaluating performance measures like as RMSE or MAE.
6) Convergence check: Sixth, subsequent to the new assessment, the algorithm ascertains if any stopping requirements are met – for example, if the enhancement in performance fails to surpass a threshold or a certain number of repetitions has been achieved.
7) Outcome: Seventh, upon satisfying the convergence conditions, the procedure concludes, yielding the optimal hyperparameter set found to date. Consequently, the loop resumes at Step 2, integrating the newly acquired data into the GP model, and a new iteration begins.

By depicting Bayesian optimization iteratively, each iteration enhances the surrogate GP model, aiming to quickly ascertain ideal hyperparameters while reducing the frequency of costly model assessments. This iterative technique is particularly advantageous for recommender systems, since hyperparameter optimization often requires substantial computation resources.

3. An Overview of Bayesian Optimization Utilizing Gaussian Processes

Assume that one has some function f that maps a set of hyperparameters $\mathcal{X}$ into a real value. However, there are instances when optimizing f is a daunting task, especially when function evaluations consume lots of time. In such situations, Bayesian optimization is a viable alternative. A detailed explanation of the basic principles underpinning Bayesian optimization can be found in [32, 33], below we summarize this explanation:

- The unknown function f is modeled as a probability distribution over functions. Probabilistically, this allows us to include prior knowledge about f. Usually, a Gaussian process serves as the commonest approach to use for this purpose.
- By evaluating the function at a few selected points $x_1, x_2, \ldots, x_D$, we get corresponding values $f(x_1), f(x_2), \ldots, f(x_D)$ which are regarded as observed data within the probabilistic model.
- By modeling f as a probability distribution, we may compute the conditional probabilities of $f(x)$ given observed function values at points $x_1, x_2, \ldots, x_D$ for example. Hence, in terms of computational costs, using a Gaussian process for this probabilistic model makes it cheaper to calculate $f(x)$ than to evaluate directly some of the desired points.
- The conditional distribution can assist in producing estimates of $f(x)$ at unseen coordinates while simultaneously guiding future choices during the optimization procedure.
- The most promising point for subsequent evaluation is determined by an acquisition function. This function relies on $f(x)$'s predicted mean and standard deviation, as well as y_{best}, which is presently the highest (or "best") observed value. Common acquisition functions include:

  1) Probability of improvement (PI): This function ranks points that have good chances of being above the current best. It emphasizes prioritizing points with a greater likelihood of enhancing the present optimal value. Demands low computation but may exhibit excessive caution in using established advantageous areas and neglect to investigate adequately when the likelihood of progress in uncharted territories seems minimal [34–36].
  2) Expected improvement (EI): This function works to maximize the possibility of improving the optimal value. It balances exploitation and exploration by assigning weights to each prospective enhancement based on its likelihood; often demonstrates effective performance in practice. It may incur further processing costs compared to PI when assessing the anticipated benefit at each potential location [34, 35].
  3) Lower confidence bound (LCB): It considers both mean and uncertainty thereby balancing exploration with exploitation. It directly regulates exploration by imposing more penalties on high-variability areas; straightforward formulation integrating mean and uncertainty. Its reactivity to the selected confidence parameter may result in excessively cautious or too aggressive searches, affecting convergence rates [34, 37].

- The primary advantage of Bayesian optimization is its ability to utilize all data available. It differs from methods that restrict themselves solely to recent or historical information by providing that each subse-

quent query incorporates every other one that occurred earlier. This makes it more efficient and accurate.

Finally, CF models' hyperparameter tuning using Bayesian optimization with GPs provides a solid and efficient framework. It typically balances exploration and exploitation through the use of probabilistic modeling as well as acquisition functions delivering more accurate and effective optimization. This means that it is an appropriate method for complex models that include high computational costs (expensive) while also enabling tracking of uncertainty that could help improve the overall performance of recommendation systems.

More information on the datasets under consideration will be provided in the next section. This will be followed by two well-known algorithms used in recommender systems and how they have been applied in our work to address the cold-start problem, limitations of using GP models for hyperparameter optimizations and sparsity in datasets.

IV. Experiments

1. Baselines and Rationale

To comprehensively assess the efficacy of our proposed method, we juxtapose it with many cutting-edge algorithms, the majority of which use neural network topologies. Tables I and II encapsulate these models, including Neural Collaborative Filtering, deep factorization machines, and several other neural-based methodologies. We have furthermore included a traditional but highly competitive baseline, SVD++, recognized for using both explicit and implicit input to enhance user preference modeling [9, 10].

The selection of these baselines fulfills many objectives:

1) Current trends representation: Neural-based algorithms have established themselves as standard in the realm of recommendations. Models like Neural CF use deep neural networks to comprehend intricate user-item interactions that conventional methods may neglect.
2) Comparative analysis: The inclusion of SVD++ highlights the disparities in performance between solely factorization-based approaches, which include minor changes to account for implicit feedback, and more contemporary neural solutions. This enables readers to discern if performance improvements arise from more complex structures, superior hyperparameter optimization, varied regularization techniques, or a confluence of variables.
3) Empirical significance: Research and contests (e.g., Netflix Prize) continuously underscore SVD++ and neural-based models as formidable challengers, with documentation from many sources (for instance, [8]) routinely attesting to their dependability and prevalence. This guarantees that the benchmarks represent acknowledged, real-world best practices.
4) Contextualizing our findings: By doing a comprehensive comparison – from traditional factorization to contemporary neural benchmarks – we provide a clear perspective on relative advantages and disadvantages, including tuning overhead, and scalability considerations. This thorough assessment offers an enhanced comprehension of how our methodology aligns with prominent and extensively used recommender system frameworks.

These baselines not only demonstrate the benefits and drawbacks of our strategy but also situate our study within the larger context of current collaborative filtering research.

2. The Datasets

The Netflix Prize competition included evaluating algorithms using an extensive dataset of over 100 million movie ratings submitted by anonymous consumers. Additionally, an extra dataset known as the Quiz set comprises 1.4 million current ratings for interim evaluation. A withheld Test set of comparable size was ultimately used to determine the competition's winner. This work uses identical datasets for training, testing, and validation, in contrast to other studies [38, 39] that adopt a modified approach by binarizing ratings and filtering users, or by randomly partitioning the original training set to generate supplementary datasets [40]. The MovieLens 1M and 10M datasets are also widely used for assessing the efficacy of recommender systems. For these two datasets, as well as the Jester dataset 2, we randomly partitioned the data in a 90:10 ratio for training and testing, a percentage often used in research, as referenced in [8]. For the Douban dataset, we use the preprocessed subsets and splits the same as the authors of [41] used. The density of the Netflix Prize, MovieLens 1M, MovieLens 10M, Jester and Douban datasets are 0.0118, 0.0447, 0.0131, 0.2595 and 0.0152 accordingly.

3. Two CF Algorithms

The two basic CF algorithms from [10] provided in this section serve as the basis for more elaborate models. In the first algorithm, users and items are embedded into a shared latent factor space. The idea is to obtain underlying preferences by mapping the users and the items on corresponding vectors. Specifically, every user u corresponds to a vector $\mathbf{p}_u \in \mathbb{R}^n$ called user-factor vector, whereas each item i correlates with a vector $\mathbf{q}_i \in \mathbb{R}^n$ named item-factor vector. The predicted ratings denoted as $\hat{r}_{ui}$ are calculated as the inner product of these vectors, namely $\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i$.

Optimization of the latent factors aims at minimizing the difference between predicted ratings and actual ratings r_{ui} during the process:

minimize $\sum_{(u,i)\in\mathcal{K}} \left(r_{ui} - \mathbf{p}_u^T \mathbf{q}_i\right)^2 + \lambda \left(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2\right)$.

$\mathcal{K}$ is a set defined as $\mathcal{K} = \{(u,i) \mid r_{ui} \text{ is known}\}$; λ is a regularization parameter that has to be tuned. Thus the prediction error for any estimate can be given as $e_{ui} = r_{ui} - \hat{r}_{ui}$. We take each training instance and pass through all ratings included in the set $\mathcal{K}$. For every rating r_{ui}, we make changes to model parameters counter to gradients given by;

- $\mathbf{p}_u \leftarrow \mathbf{p}_u + l_r \cdot (e_{ui} \cdot \mathbf{q}_i - \lambda \cdot \mathbf{p}_u)$
- $\mathbf{q}_i \leftarrow \mathbf{q}_i + l_r \cdot (e_{ui} \cdot \mathbf{p}_u - \lambda \cdot \mathbf{q}_i)$

where l_r is the learning rate.

The second CF algorithm builds on top of the first CF algorithm by adding baseline estimation for handling the underlying user and item biases present in CF data. These biases occur as users or items with systematic tendencies toward higher or lower ratings. To tackle these problems, baseline estimates are utilized: μ symbolizes the general mean rating while b_u and b_i represent user and item deviations respectively from the unbiased part $\mu + \mathbf{p}_u^T \mathbf{q}_i$. The prediction can then be computed as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^T \mathbf{q}_i$$

The associated loss is minimized to estimate parameters:

$$\text{minimize} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i - \mathbf{p}_u^T \mathbf{q}_i)^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2)$$

Similar to the first algorithm, we loop over all known ratings in the data set $\mathcal{K}$. For every rating r_{ui}, model parameters are updated like this:

- $b_u \leftarrow b_u + l_r \cdot (e_{ui} - \lambda \cdot b_u)$
- $b_i \leftarrow b_i + l_r \cdot (e_{ui} - \lambda \cdot b_i)$
- $\mathbf{p}_u \leftarrow \mathbf{p}_u + l_r \cdot (e_{ui} \cdot \mathbf{q}_i - \lambda \cdot \mathbf{p}_u)$
- $\mathbf{q}_i \leftarrow \mathbf{q}_i + l_r \cdot (e_{ui} \cdot \mathbf{p}_u - \lambda \cdot \mathbf{q}_i)$

The algorithms iterate through $\mathcal{K}$ many times and terminate when the loss fails to decrease significantly after multiple iterations.

4. Limitations in GP Models for Hyperparameter Tuning

While GP models are helpful in balancing exploration and exploitation during hyperparameter optimization, we recognize many limitations intrinsic to this methodology. Initially, GP-based methodologies may encounter scaling challenges in high-dimensional parameter spaces, since matrix operations often become computationally unfeasible [29, 30]. The smoothness assumption in GPs may be too limiting for some CF algorithms, especially those exhibiting significant performance declines due to tiny hyperparameter adjustments [29]. Third, the optimization process is significantly affected by initial hyperparameter samples, and inadequate seeding may confine the search to local minima [42].

This work prioritized practical constraints – specifically time and computational resources – over a comprehensive exploration of advanced GP modifications (e.g., sparse approximations or customized GP models) that may alleviate some of these issues. Our future research agenda includes the investigation of scalable variations of GP models designed for high dimensions; alternative GP models that are more appropriate for sudden fluctuations in CF performance, and enhanced methods for initializing GP hyperparameter searches, particularly under cold-start scenarios.

Investigating these pathways may enhance the efficacy of GP-based Bayesian optimization in bigger and more complex CF contexts. Nonetheless, implementing these improvements requires substantial further investment in methodological advancement and computer capacity, which we will address in further research.

5. The Cold-Start Problem

The accuracy of recommendations and user contentment in collaborative filtering systems are particularly susceptible to cold-start scenarios, characterized by a lack of previous data for new users or products. Numerous sophisticated techniques, including meta-learning, deep learning and hybrid approaches [43–47] have been suggested to tackle this issue. Nevertheless, these methodologies often need considerable data and computing resources [48, 49].

Two CF algorithms used in this research utilize global averages to address cold-start cases during validation and testing. This strategy offers a rapid and computationally economical baseline, although it unavoidably reduces the intrinsic complexity of varied user behavior [44, 45]. This dependence on global averages neglects individual- or item-specific intricacies and may generate bias by presuming uniform user behavior, ignoring personal preferences. This limitation belongs to these two CF algorithms and not to the proposed technique in this research.

Owing to practical limitations – including restricted time, computer resources, and research scope – we have not explored more sophisticated cold-start alternatives. Our primary objective is to provide an efficient method instead of an all-encompassing solution that addresses every facet of the cold-start issue. Future study will investigate hybrid or meta-learning methodologies to enhance the personalization

of early-stage suggestions for new users and enhanced user/item initialization techniques that surpass global averages.

In conclusion, we recognize that the cold-start approach used in the two CF algorithms serves as an initial framework – it sacrifices detail for operational efficiency. Significant progress is achievable and will be sought in future endeavors as resources and project schedules permit.

6. Sparse Data Issues

This subsection outlines the impact of data sparsity – a prevalent issue in CF – on recommendation systems.

Data sparsity impact: this issue is considerably prominent in CF, impacting the stability of user-item interaction matrices with scarce data. With limited data points, there is not enough data that may accurately portray underlying preferences and latent characteristics. Consequently, this gives larger prediction errors with higher RMSE values, as shown by comparison tests for traditional and neural modes [50, 51].

Algorithm adapability: traditional means, for instance, the SVD++ method, otherwise manage sparsity well by integrating both the implicit and data inputs. However, with the most severe scarcity of user-item interactions, the predictive capability of these methods will similarly be significantly reduced [50]. Gaussian priors in the Bayesian hyperparameter tuning may improve estimates of their parameters, though the inherent scarcity of data will be blocking an array of possible applications [51].

Error propagation in latent factor models: when data sparsity prevents the learning of complex user-item interactions, the resultant prediction errors will propagate via the neural network. Deep learning methods like Neural Collaborative Filtering were designed to recognize more subtle patterns, but the advantage of these sophisticated models is not fully employed in sparse datasets, creating a mismatch between the theoretical potential and actual usefulness of the model [52].

In conclusion, sparse interaction matrices diminish the efficacy of even cutting-edge CF methods by hindering the learning process for latent user-item interactions. This leads to an increased RMSE and decreased suggestion accuracy.

The results of Bayesian optimization using Gaussian processes to choose the hyperparameters of the two methods on the experiment datasets will be shown in the next section.

V. Results

1. Experimental Setup and Results

In our experiments, this work used scikit-optimize [53] for hyperparameter optimization, which implements

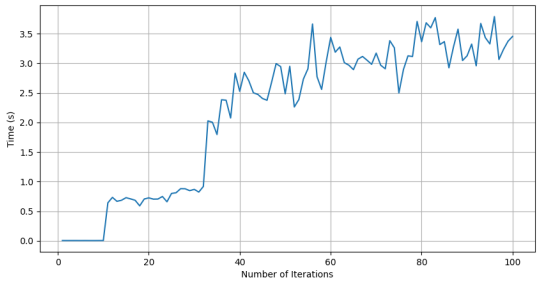


Figure 2. Bayesian optimization time.

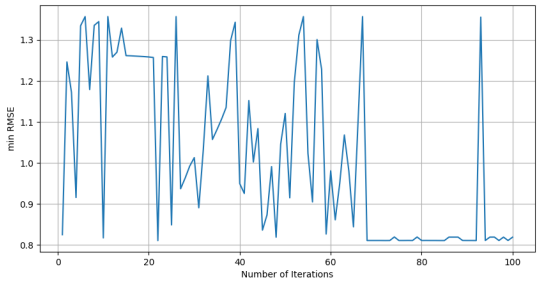


Figure 3. RMSE convergence of the first algorithm on the Netflix Prize dataset.

Bayesian optimization. Specifically, we use its gp_minimize function with default values for all possible parameters. Especially, we use the "gp_hedge" value (the default value) for the acquisition function parameter. This value will direct the gp_minimize function to probabilistically choose one of the three acquisition functions (e.g., PI, EI, LCB) at every iteration based on the gain provided by each function, with the trade-off being variability in convergence rates. Note that in this research, we do not try to achieve state-of-the-art performance as well as the effect of the chosen acquisition function on the performance of CF algorithms; instead, we focus on practical usages of Bayesian optimization with Gaussian priors in recommendation systems by experiment-

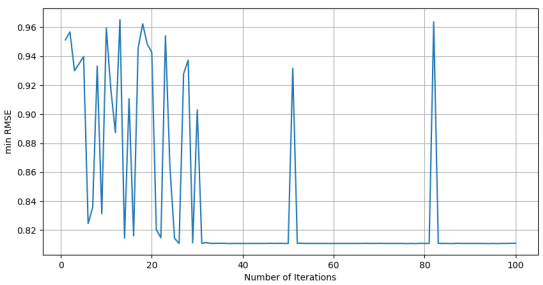


Figure 4. RMSE convergence of the second algorithm on the Netflix Prize dataset.

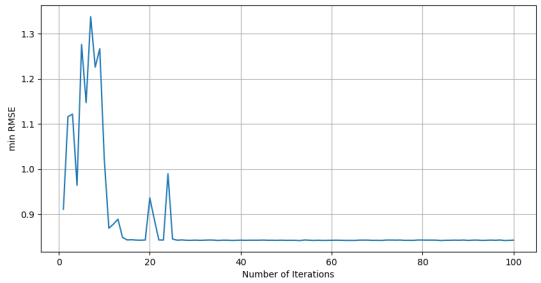


Figure 5. RMSE convergence of the first algorithm on the MovieLens 1M dataset.

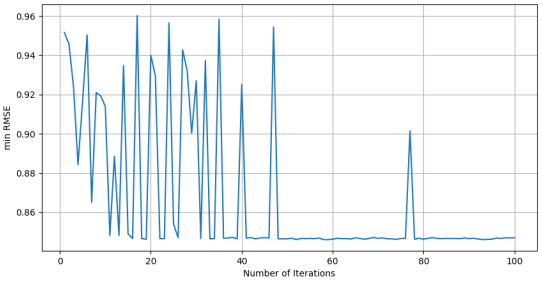


Figure 6. RMSE convergence of the second algorithm on the MovieLens 1M dataset.

ing with two simple and basic CF algorithms and some popular datasets. This aspect also signifies the simplicity of deployment and integration of the suggested approach into real-world systems.

In the experiments, we set the initial learning rate l_r to 5e-3, the range of the regularization parameter λ from 1e-4 to 1.0, and the dimension n range from 50 to 500. If the loss reduction is less than 1e-4 after two iterations, the learning rate is reduced by 10 times. The parameter optimization process, which optimizes $b_u, b_i, \mathbf{p}_u, \mathbf{q}_i$, stops if the loss reduction remains below 1e-4 after five iterations (stopping criteria).

Table I
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS ON THE MOVIELENS 1M DATASET.

Algorithm	**RMSE (MovieLens 1M)**
LLORMA	0.833
I-AutoRec	0.831
CF-NADE	0.829
GC-MC	0.832
GraphRec	0.843
GraphRec+Extra	0.842
SparseFC	0.824
IGMC	0.857
GLocal-K	0.822
Our approach (first algo.)	**0.8410**
Our approach (second algo.)	**0.8459**

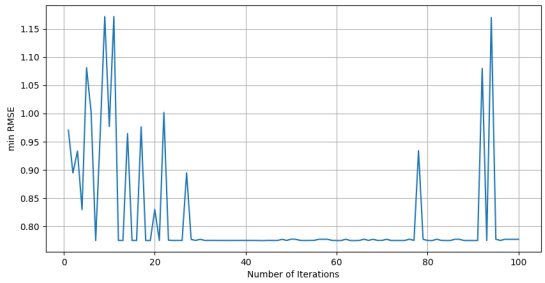


Figure 7. RMSE convergence of the first algorithm on the MovieLens 10M dataset.

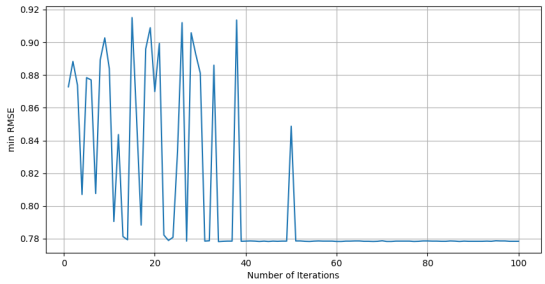


Figure 8. RMSE convergence of the second algorithm on the MovieLens 10M dataset.

Table II
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS ON THE MOVIELENS 10M DATASET.

Algorithm	**RMSE (MovieLens 10M)**
RSVD	0.8256
U-RBM	0.823
BPMF	0.8197
APG	0.8101
DFC	0.8067
Biased MF	0.803
GSMF	0.8012
SVDFeature	0.7907
ALS-WR	0.7830
I-AutoRec	0.782
LLORMA	0.7815
WEMAREC	0.7769
I-CFN++	0.7754
MPMA	0.7712
CF-NADE 2 layers	0.771
SMA	0.7682
GLOMA	0.7672
ERMMA	0.7670
AdaError	0.7644
MRMA	0.7634
SGD MF	0.7720
Bayesian MF	0.7633
Bayesian timeSVD	0.7587
Bayesian SVD++	0.7563
Bayesian timeSVD++	0.7523
Bayesian timeSVD++ flipped	0.7485
Our approach (first algo.)	**0.7757**
Our approach (second algo.)	**0.7787**

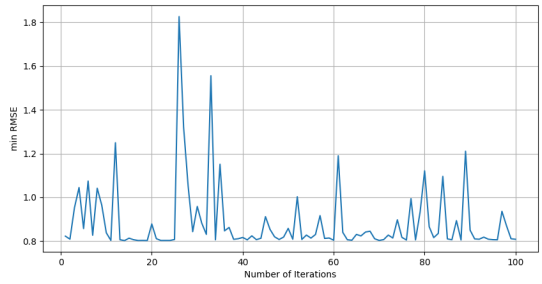


Figure 9. RMSE convergence of the first algorithm on the Douban dataset.

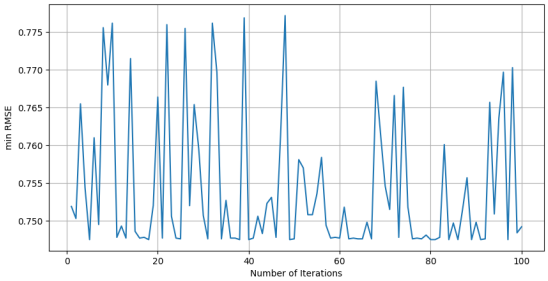


Figure 10. RMSE convergence of the second algorithm on the Douban dataset

Table III
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS ON THE DOUBAN DATASET.

Algorithm	RMSE (Douban)
GC-MC+Extra	0.734
GRAEM	0.732
SparseFC	0.730
IGMC	0.721
MG-GAT	0.727
GLocal-K	0.721
Our approach (first algo.)	**0.8055**
Our approach (second algo.)	**0.7499**

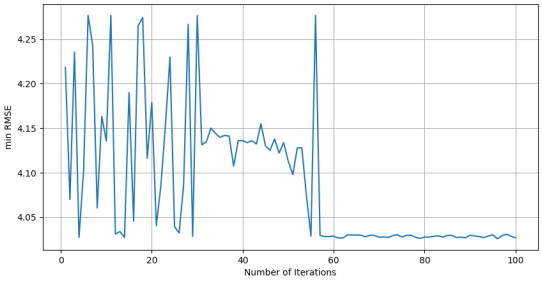


Figure 11. RMSE convergence of the first algorithm on the Jester dataset 2.

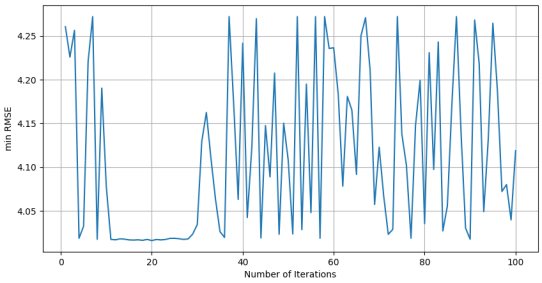


Figure 12. RMSE convergence of the second algorithm on the Jester dataset 2.

Table IV
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS ON THE JESTER DATASET 2.

Algorithm	RMSE (Jester)
SVD	4.497
SVD++	4.904
NMF	6.324
Slope One	4.253
Co-Clustering	4.292
Our approach (first algo.)	**4.036**
Our approach (second algo.)	**4.025**

Figures 3 and 4 illustrate the convergence of the minimum RMSE value after 100 iterations with the two algorithms, which achieved an RMSE of 0.8103 (λ = 3.36E-02 and n = 90) and 0.8107 (λ = 0.026905275 and n = 87) on the Test set of the Netflix Prize datasets accordingly. As mentioned before, recent research works [38, 39] use a modified version of the datasets by binarizing ratings and filtering users, or randomly splitting the original training set to create additional datasets [40]. This prevents us from comparing the results, but the best-performing algorithm in the Netflix Prize contest had achieved an RMSE score of 0.8567 on the Test set [9], which incorporated SVD++ and many other techniques.

Table I, II, III and IV show the results from the two CF algorithms that this work used to calculate the errors on MovieLens 1M, MovieLens 10M, Douban and Jester datasets. For the MovieLens 1M dataset, other algorithms' performances are taken from [41] while those for the MovieLens 10M and Douban datasets are available from [8]. For the Jester dataset 2, we ran all possible algorithms on our computer using the scikit-surprise toolkit [54] because we could not find state-of-the-art results for this particular dataset. On the whole, the outcome is an indication that in hyperparameter tuning, Bayesian optimization using Gaussian priors is effective for both CF models.

2. Result Analysis

a) Factors Affecting Algorithm Performance

Most recommender algorithms are capable of high diversity concerning the nature of the datasets used for testing, thus letting several other factors become significant contributors to the reasons for the superiority of an algorithm over the lesser-performing ones:

- Sparsity and density of the dataset: the datasets vary to an enormous degree between sparsity. If we persist in calling MovieLens 10M dataset having higher sparsity (density = 0.0131) compared to Jester (more dense, with a density to approximate 0.2595), the algorithm using implicit feedback must handle this sparsity consideration. SVD++ or its variations, for instance, provide a good means for handling sparsity because they use both types of inputs, implicit and explicit, describing user preference in greater detail for opting for their sparser dataset. However, with denser datasets, simpler CF approaches – especially performance-optimized for sparseness – might be enough to predict user-item interactions with competitive error rates.
- Model complexity and feedback handling: advanced algorithms, mostly those based on neural networks (e.g., Neural Collaborative Filtering), are inherently very good at investigating intricate relationships between users and products. Yet, they hardly lead to significant performance improvements over traditional techniques when user interactions are scarce in the dataset. Both Netflix Prize and MovieLens 1M datasets, independently, could not raise a strong lead for neural management since the best-performing classical CF algorithms optimized via Bayesian optimization could reach RMSE levels that were competitive with these models. It, therefore, suggests that increasing model complexity using neural methods was required to model user-item interactions in the form of subtle patterns, often missed by less complex techniques.
- Hyperparameter adjustment and optimization techniques: hyperparameter tuning is a crucial step. The research revealed that using Bayesian optimization with GPs may efficiently fine-tune even fundamental CF methods. This is especially advantageous in practical situations where manual adjustment is laborious. The automatic exploration of hyperparameter space enables the model to adjust to the distinct noise and user-item interactions of each dataset. For instance, adjusting parameters allowed the simple CF models to exhibit strong performance across various datasets. The improvement in RMSE scores – exemplified by a result of 0.8103 on the Netflix Prize Test set – demonstrates the algorithm's capacity for self-adjustment based on dataset features.

The differentiation in overall performance of algorithms is based largely on the adequacy of the algorithm design to cope with implying sparsity, noise, and density of various datasets, and the inherent informative signaling using explicit and implicit feedback. This indicates that if algorithms offer mechanisms for efficient hyperparameter tuning – such as Bayesian optimization employing Gaussian processes – they may be rendered resilient to the specific obstacles presented by each dataset.

b) Emerging Trends in Algorithm Convergence and Performance

Several noteworthy features in convergence behavior and overall performance are demonstrated by the experimental results:

- Iterative convergence behavior:
  1) Through the use of GPs, the Bayesian optimization method propels the hyperparameters' progressive convergence across iterations. Up until a plateau is reached, the RMSE steadily improves as each iteration improves the model's predictions by updating the GP with fresh performance evaluations.
  2) In the studies that were reported, convergence was often seen in 50–90 repetitions. This shows that the hyperparameter tweaking rapidly focuses on efficient areas of the search space, even with the inherent unpredictability brought about by various acquisition functions.
  3) Convergence curves, which are shown in many figures, usually demonstrate a slow decline in RMSE over time. Both algorithms rapidly approach a comparable RMSE for some datasets, like the Netflix Prize dataset, indicating that even little hyperparameter changes can have a big effect when the tuning procedure is well calibrated.
- Performance trends across datasets:
  1) The performance results show competitive RMSE values across a variety of datasets, including Netflix Prize, MovieLens 1M, MovieLens 10M, Douban, and Jester.
  2) The main objective of this research work was to show the practical usability of Bayesian optimization in recommender systems. On the Netflix Prize dataset, both straightforward CF approaches tuned via Bayesian optimization achieved RMSE values around 0.81. This suggests that automated hyperparameter tuning

achieves competitive performance with a significant reduction in manual work.

3) The tuned models achieved RMSE levels in the range of 0.775 to 0.779 for bigger and more intricate datasets, such as MovieLens 10M. The studies demonstrate that even simple CF models gain a great deal from the optimization process, even though more complex approaches in the literature occasionally produce lower RMSE values.
4) The two algorithms' outputs differed more notably in the Douban and Jester datasets. For instance, both algorithms did well on the Jester dataset (RMSE approximately 4.02–4.04), but the second algorithm significantly improved on the Douban dataset (RMSE dropping from 0.8055 to 0.7499). This variety illustrates the ways in which dataset properties, including density, noise, and sparsity, impact convergence dynamics and overall performance.

In conclusion, the performance trends across datasets show that even simple CF algorithms can produce competitive results when combined with Bayesian optimization for hyperparameter tuning, while the convergence trends show an efficient iterative tuning process with quick progress in lowering RMSE. The importance of incorporating automated optimization techniques into recommender systems is highlighted by this convergence–performance link.

3. Scalability Analysis

a) Computational Impact of Bayesian Optimization with GPs

The complexity of Bayesian optimization with GPs is mainly defined by two tasks at each step: tuning Gaussian process hyperparameters and maximizing the acquisition function for new hyperparameter selection. The complexity of each task incur $O(\mathbb{N}^3)$ and $O(\mathbb{N}^2)$ computational cost respectively, where N is the number of data points [55]. Note that the optimization of acquisition functions (e.g., PI, EI, UCB) generally scales with the dimensionality d of the search space (not the size or the number of samples of datasets with which we conduct recommendations). For fixed or small values of d, this can be viewed as a lower-order term (e.g., $O(\mathbb{N}^2)$ in relation to GP inference [56]), and this characteristic is also mentioned in [12].

Our experience and results show that when you optimize parameters over one iteration for instance, it requires much more time than one iteration of the Bayesian optimization technique, hence the time taken searching for a new set of hyperparameters (a new exploration point) is almost negligible compared to the time taken while optimizing parameters for CF algorithm. Figure 2 displays the time taken to conduct our experiment on Bayesian optimization with Gaussian processes having two hyperparameters and lasting up to about 100 iterations. Instead, this time consumption is not influenced by the characteristics of the CF algorithms since the CF algorithms operate blindly within the Bayesian optimization process.

b) Scalability Strategies

While our present study does not examine parallelization or deployment strategies, we acknowledge their significance for large-scale implementations. Prospective solutions (e.g., Spark MLlib for Collaborative Filtering [57], containerized microservices, real-time streaming, and autoscaling infrastructure) signifies a future trajectory in enhancing our recommendation pipeline for more extensive, resource-demanding environments:

Parallelization Strategies

- Distributed matrix factorization: frameworks like Spark MLlib for Collaborative Filtering enhance the efficiency and memory management of large-scale factorization by distributing the user–item matrix over numerous nodes.
- Memory-based CF and partitioning: distributing users and/or items among computers facilitates partial similarity computations, which are then combined to provide comprehensive suggestions.

Deployment Approaches

- Hybrid batch and online learning: a dual-tier framework (bulk training combined with incremental updates) enables the continuous integration of fresh data without interrupting real-time predictions.
- Autoscaling infrastructure: cloud systems automatically augment computational resources as needed, hence ensuring low-latency suggestions despite a growth in users or requests.

4. Hyperparameter Analysis

In the experiment, we selected the hyperparameter ranges based on past experience; we made them quite wide so that optimizations could be made on them more freely (λ and n), and the starting value of the learning rate l_r. Broader search spaces enhance model capacity by allowing multiple aspects of the objective function to be explored, thus potentially improving locate optimal solutions; however, numerous evaluations and more complex management of exploration-exploitation trade-offs are required by such an expansion [58, 59]. In other words, most hyperparameter ranges are normally determined using known techniques

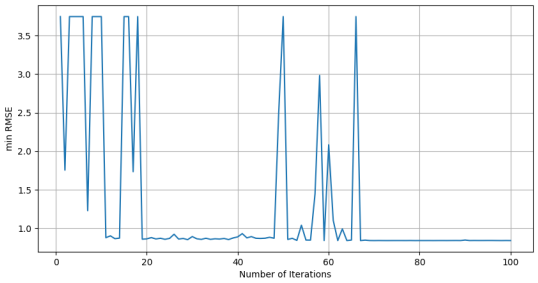


Figure 13. RMSE convergence of the first algorithm on the MovieLens 1M dataset (with hyperparameter space broaden).

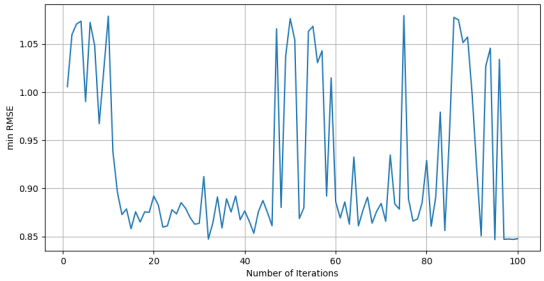


Figure 14. RMSE convergence of the second algorithm on the MovieLens 1M dataset (with hyperparameter space broaden).

or by observing various patterns (empirical data) observed over time. Secondly, but just like other previous research outcomes [11–13], it was found that the hyperparameter tuning process converges within 50–90 iterations. Lastly, all methods we used have only two hyperparameters in our experiment; however, using more than this number increases the cost of Bayesian optimization with Gaussian processes; though this cost does not considerably rise with the number of parameters as reported in [12, 60, 61].

5. Robustness Analyses

In this section, we present robustness analyses to investigate how robust and fine the method is. We delve more into the effect of hyperparameter changes and dataset variables on the performance of both CF methods.

a) Hyperparameter Sensitivity Analysis

In this analysis, we broaden the range of the two hyperparameters (λ and n) used in the Bayesian optimization process for the MovieLens 1M and 10M, Douban and Jester datasets (due to time and physical resource constraints, we could not be able to experiment further with the Netflix Prize dataset because of its size). This will allow us to empirically assess the convergence behavior and robustness of the proposed technique. This facilitates the examination of a broader spectrum of parameter values, possibly uncovering new optimum areas and elucidating the sensitivity and resilience of each method across diverse hyperparameter configurations.

In this new experiment, we set the range of the regularization parameter λ from 1e-6 to 10, and the dimension n ranges from 10 to 1000. The RMSE convergence of the two algorithms for the datasets are presented from Figure 13 to Figure 20. The performance of the two CF algorithms is presented in Table V.

The new figures indicate that a larger optimization area results in increased variability in convergence/performance trends. Certain hyperparameters, when let to fluctuate

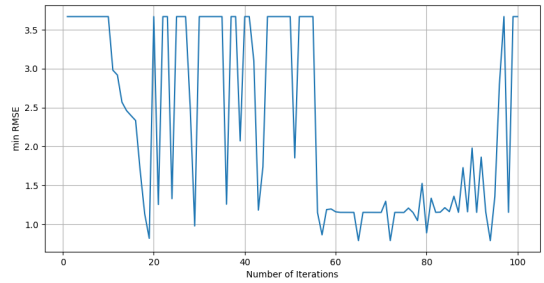


Figure 15. RMSE convergence of the first algorithm on the MovieLens 10M dataset (with hyperparameter space broaden).

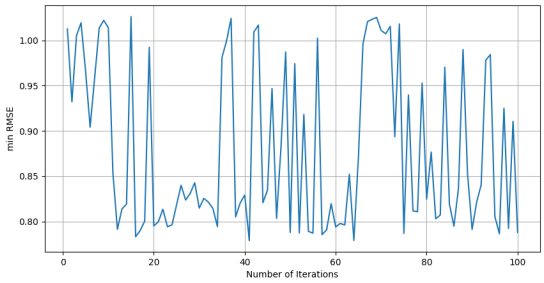


Figure 16. RMSE convergence of the second algorithm on the MovieLens 10M dataset (with hyperparameter space broaden)

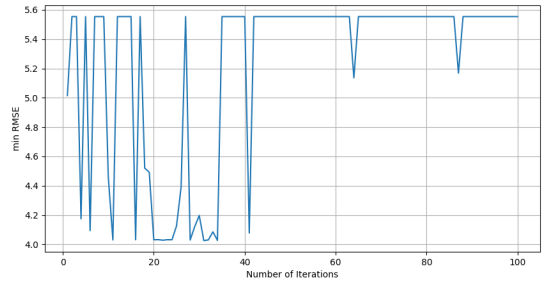


Figure 17. RMSE convergence of the first algorithm on the Jester dataset 2 (with hyperparameter space broaden).

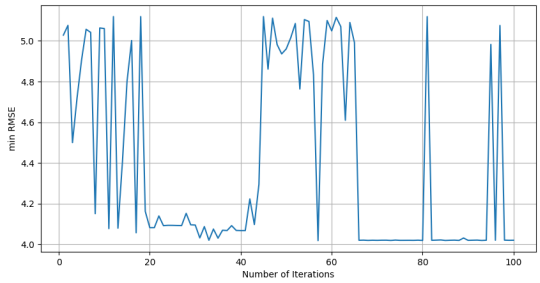


Figure 18. RMSE convergence of the second algorithm on the Jester dataset 2 (with hyperparameter space broaden).

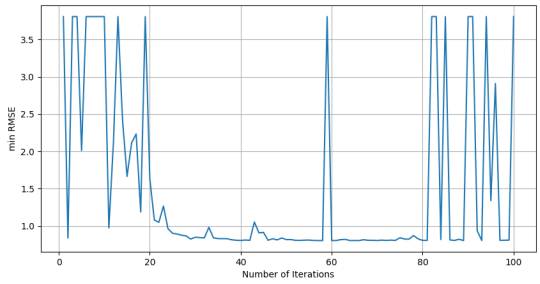


Figure 19. RMSE convergence of the first algorithm on the Douban dataset (with hyperparameter space broaden).

within a wider spectrum, lead to more significant alterations in RMSE, underscoring essential thresholds beyond which performance deteriorates. The previous, tightly defined space (Figures 5 to Figure 12) validated the baseline performance (”baseline“ means that is based on experience) of both methods, but the expanded optimization space (Figure 13 to Figure 20) exposes trade-offs. Specific hyperparameters may provide advantages just within a restricted domain, and broadening the scope might reveal weaknesses – such as areas where the model exhibits instability or overfitting. This contrast highlights the need of carefully balancing exploration and exploitation in hyperparameter optimization. The findings suggest that while fine-tuning within a limited scope might provide satisfactory performance, an expansive search may reveal additional ideal configurations that could further decrease RMSE. Nonetheless, this incurs the expense of heightened sensitivity in other instances, especially for the initial algorithm.

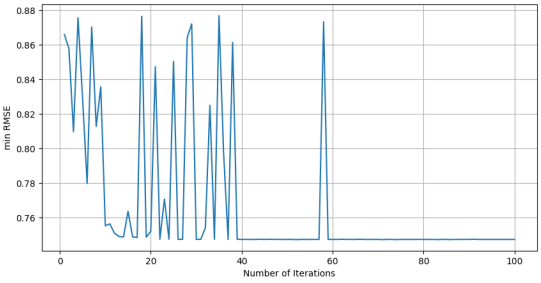


Figure 20. RMSE convergence of the second algorithm on the Douban dataset (with hyperparameter space broaden).

Table V
PERFORMANCE OF THE TWO CF MODELS ON THE EXPERIMENTAL DATASETS WITH THE HYPERPARAMETER SPACE BROADEN.

Algorithm	MovieLens 1M	MovieLens 10M	Jester	Douban
First algo.	0.8409	0.7917	4.034	0.8072
Second algo.	0.8461	0.7791	4.025	0.7498

For the minimum RMSE achieved in the two hyperparameter search spaces, both algorithms yielded generally stable RMSE values; however, these values sometimes plateaued at suboptimal levels owing to the constrained search range.

b) Robustness Across Datasets

This subsection summarizes the results of our tests in the datasets of the experiment.

Netflix Prize dataset: our experiments show that both CF models, when calibrated via Bayesian optimization, quickly converge toward competitive error rates. Specifically, the first algorithm achieved an RMSE of 0.8103, the second algorithm achieved a very similar performance with an RMSE of 0.8107. These results indicate that even with relatively simple CF architectures, automated hyperparameter tuning can yield results that are on par with more complex methods.

MovieLens 1M dataset: the two algorithms got the RMSE values of 0.8410 and 0.8459 accordingly. While a few state-of-the-art techniques typically achieve marginally lower values on this dataset, our methods demonstrate that with decreased manual intervention, competitive performance is attainable.

MovieLens 10M dataset: on the larger MovieLens 10M dataset –characterized by a lower density (0.0131) – our methods again proved effective with the RMSE values of 0.7757 and 0.7787 accordingly. These results emphasize that Bayesian hyperparameter search allows even straightforward CF models to adapt efficiently to larger and more sparse datasets, narrowing the performance gap with more intricate network-based approaches.

Douban dataset: for the Douban dataset, which poses additional challenges due to its user-item interactions and sparsity, our study shows the RMSE values of 0.8055 and 0.7499 accordingly. The significant performance improvement observed with the second algorithm on this dataset highlights how modeling strategies can be tailored to better

mitigate the adverse effects associated with higher data sparsity.

Jester Dataset: finally, in experiments conducted on the Jester dataset 2, we employed scikit-surprise for a consistent evaluation framework. The RMSE results are 4.036 and 4.025. On this denser dataset, both algorithms achieved very similar error rates, supporting the claim that our Bayesian optimization approach is robust across various data distributions.

Overall observations on convergence and robustness: across all datasets in our experiment, convergence curves (refer to Figures 3–12 in Section V.1) reveal rapid initial improvement, with diminishing returns after approximately 50–90 iterations – indicative of effective exploration within the hyperparameter space.

6. Practical Implications

Our work demonstrates that Bayesian optimization with Gaussian processes for tuning hyperparameters of two basic and simple CF algorithms on three popular datasets yields very competitive performance. We think that different issue sets dominated by each approach account for this enhancement. Although there are situations where minor improvements may occur if you switch algorithms in one approach, it is more efficient to use a combination of several approaches aimed at solving specific sets of issues than making changes within one approach; this way many different approaches could be developed to solve various issues that influence CF models. In terms of return on investment, reliable algorithms are preferable to sophisticate yet less reliable ones, according to [62–64]. Practitioners should choose more advanced algorithms over simplistic CF ones if they want better results; however, they must make sure that these other properties are balanced such as complexity, usability, and stability so that it will not become less efficient after all.

VI. Conclusion

This research analyzed the efficacy of two fundamental CF methods augmented by Bayesian optimization using GPs for automated hyperparameter tuning. Our experimental investigation of datasets including Netflix Prize, MovieLens (1M and 10M), Jester, and Douban has shown that even basic CF models may attain competitive performance with meticulous hyperparameter tuning, therefore reducing the need on human adjustments.

This study significantly contributes by explicitly demonstrating how Bayesian optimization systematically investigates the hyperparameter space, resulting in enhanced performance efficiency. As detailed in Section IV, our approach utilizes the iterative characteristics of GPs to achieve optimum configurations, thereby alleviating problems like data sparsity and model underfitting. The findings – exemplified by an RMSE of 0.8103 on the Netflix Prize Test set – highlight the efficacy of this method in improving model precision while conserving time and processing resources.

Moreover, the work has identified other obstacles that persist for future investigation. Our experimental setup tackled the cold-start issue using a global-averages strategy for new users and items; nonetheless, this approach is acknowledged as a baseline that reduces the intricacies of actual user behavior. As outlined in Section IV.5, advanced strategies – possibly using hybrid or meta-learning methodologies – may provide a more tailored initialization that reconciles efficiency with precision.

Likewise, the challenges posed by data sparsity (elaborated in Section IV.6) underscore that even cutting-edge optimization methods may not entirely surmount the inherent constraints of sparse interaction matrices. Future research should explore the implementation of scalable GP variations and sophisticated partitioning or distributed matrix factorization techniques (e.g., using frameworks such as Spark MLlib) to mitigate these limitations in bigger and more dynamic settings.

In conclusion, the article demonstrates that automated hyperparameter tweaking using Bayesian optimization utilizing GP models may substantially improve the efficacy of CF algorithms. The effectiveness of this method is apparent across several datasets; nonetheless, enhancements are required, especially concerning the cold-start issue and data sparsity. An explicit, cohesive narrative from the literature review to the experimental findings underscores the promise of integrating conventional CF approaches with contemporary optimization tools, paving the way for future research initiatives.

This paper fully links our methodology, experimental results, and theoretical insights, offering a comprehensive assessment of existing methodologies and a framework for future progress in recommender systems. The iterative and modular characteristics of our methodology allow ongoing enhancements, guaranteeing that as computer resources and algorithmic complexity advance, the robustness and relevance of CF models will similarly expand in more intricate settings.

REFERENCES

[1] K. Srikumar, "A framework of agent-based personalized recommender system for e-commerce," *South Asian journal of management*, vol. 11, p. 66, 2004.

[2] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert Systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] S. Yan, "A collaborative filtering recommender approach by investigating interactions of interest and trust," in *Knowledge Engineering and Management*, Berlin, Heidelberg, 2014, pp. 173–188.

[5] C. P. Lam, "Collaborative filtering using associative neural memory," in *Intelligent Techniques for Web Personalization*, Berlin, Heidelberg, 2005, pp. 153–168.

[6] R. Zhang, Q.-d. Liu, Chun-Gui, J.-X. Wei, and Huiyi-Ma, "Collaborative filtering for recommender systems," in *2014 Second International Conference on Advanced Cloud and Big Data*, 2014, pp. 301–308.

[7] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[8] S. Rendle, L. Zhang, and Y. Koren, "On the difficulty of evaluating baselines: A study on recommender systems," *ArXiv*, vol. abs/1905.01395, 2019.

[9] Y. Koren, "The bellkor solution to the netflix grand prize," 2009.

[10] ——, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08, New York, NY, USA, 2008, p. 426–434.

[11] M. Imani, M. Imani, and S. F. Ghoreishi, "Bayesian optimization for expensive smooth-varying functions," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 44–55, 2022.

[12] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter, "Applying bayesian optimization with gaussian process regression to computational fluid dynamics problems," *Journal of Computational Physics*, vol. 449, p. 110788, Jan. 2022.

[13] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12, Red Hook, NY, USA, 2012, p. 2951–2959.

[14] S. Molaei, A. Havvaei, H. Zare, and M. Jalili, "Collaborative deep forest learning for recommender systems," *IEEE Access*, vol. 9, pp. 22 053–22 061, 2021.

[15] M. F. Aljunid and M. Doddaghatta Huchaiah, "Multi-model deep learning approach for collaborative filtering recommendation system," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 4, p. 268–275, nov 2020.

[16] Y. Huang and J. T. Kwok, "Collaborative filtering via co-factorization of individuals and groups," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–8.

[17] Z. Li, J. Huang, and N. Zhong, "Exploiting user and item embedding in latent factor models for recommendations," in *Proceedings of the International Conference on Web Intelligence*, ser. WI '17, New York, NY, USA, 2017, p. 1241–1245.

[18] F. Horasan, A. H. Yurttakal, and S. Gündüz, "A novel model based collaborative filtering recommender system via truncated ulv decomposition," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101724, 2023.

[19] Y. Chen, S. Mensah, F. Ma, H. Wang, and Z. Jiang, "Collaborative filtering grounded on knowledge graphs," *Pattern Recognition Letters*, vol. 151, pp. 55–61, 2021.

[20] N. Khaledian and F. Mardukhi, "Cfmt: a collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 5, pp. 2667–2683, May 2022.

[21] H. Xia, Y. Luo, and Y. Liu, "Attention neural collaboration filtering based on gru for recommender systems," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1367–1379, June 2021.

[22] S. Rendle, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, May 2012.

[23] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, New York, NY, USA, 2008, p. 880–887.

[24] P. Szabó and B. Genge, "Hybrid hyper-parameter optimization for collaborative filtering," in *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2020, pp. 210–217.

[25] R. Bardenet, M. Brendel, B. Kégl, and M. Sebag, "Collaborative hyperparameter tuning," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13, 2013, p. II–199–II–207.

[26] M. N. Volkovs and R. S. Zemel, "Collaborative ranking with 17 parameters," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12, Red Hook, NY, USA, 2012, p. 2294–2302.

[27] H. H. Hoos, *Automated Algorithm Configuration and Parameter Tuning*, Berlin, Heidelberg, 2012, pp. 37–71.

[28] P. N. Koch, O. Golovidov, S. Gardner, B. Wujek, J. D. Griffin, and Y. Xu, "Autotune: A derivative-free optimization framework for hyperparameter tuning," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[29] Q. Lu, K. D. Polyzos, B. Li, and G. B. Giannakis, "Surrogate modeling for bayesian optimization beyond a single gaussian process," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11 283–11 296, 2023.

[30] P. Lualdi, R. Sturm, A. Camero, and T. Siefkes, "An uncertainty-based objective function for hyperparameter optimization in gaussian processes applied to expensive black-box problems," *Applied Soft Computing*, vol. 154, p. 111325, 2024.

[31] M. A. L. Pearce, M. Poloczek, and J. Branke, "Bayesian optimization allowing for common random numbers," *Oper. Res.*, vol. 70, no. 6, p. 3457–3472, nov 2022.

[32] P. I. Frazier, "A tutorial on bayesian optimization," 2018.

[33] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[34] H. Wang, B. van Stein, M. Emmerich, and T. Back, "A new acquisition function for bayesian optimization based on the moment-generating function," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 507–512.

[35] M. Pearce, "Acquisition functions for simultaneous bayesian

optimisation of multiple problems," in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 4618–4619.
[36] J. Berk, V. Nguyen, S. Gupta, S. Rana, and S. Venkatesh, "Exploration enhanced expected improvement for bayesian optimization," in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2019, pp. 621–637.
[37] M. T. M. Emmerich, A. H. Deutz, and J. W. Klinkenberg, "Hypervolume-based expected improvement: Monotonicity properties and exact computation," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2147–2154.
[38] D. Kim and B. Suh, "Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms," in *Proceedings of the 13th ACM Conference on Recommender Systems*, ser. RecSys '19, Sep. 2019.
[39] H. Steck, "Embarrassingly shallow autoencoders for sparse data," in *The World Wide Web Conference*, ser. WWW '19, May 2019.
[40] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, "Recvae: A new variational autoencoder for top-n recommendations with implicit feedback," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, ser. WSDM '20, New York, NY, USA, 2020, p. 528–536.
[41] S. C. Han, T. Lim, S. Long, B. Burgstaller, and J. Poon, "Glocal-k: Global and local kernels for recommender systems," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21, New York, NY, USA, 2021, p. 3063–3067.
[42] Z. Chen and B. Wang, "How priors of initial hyperparameters affect gaussian process regression models," *Neurocomputing*, vol. 275, pp. 1702–1710, 2018.
[43] Y. Liu, S. Wang, X. Li, and F. Sun, "A meta-adversarial framework for cross-domain cold-start recommendation," *Data Science and Engineering*, vol. 9, no. 2, pp. 238–249, June 2024.
[44] H. Xu, C. Li, Y. Zhang, L. Duan, I. W. Tsang, and J. Shao, "Metacar: Cross-domain meta-augmentation for content-aware recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8199–8212, 2023.
[45] M. Li, W. Que, Z. Geng, M. Li, Z. Kou, J. Chen, C. Guo, and B. Zhang, "Cold-start item recommendation for representation learning based on heterogeneous information networks with fusion side information," *Future Generation Computer Systems*, vol. 149, pp. 227–239, 2023.
[46] P. Magron and C. Févotte, "Neural content-aware collaborative filtering for cold-start music recommendation," *Data Mining and Knowledge Discovery*, vol. 36, no. 5, pp. 1971–2005, September 2022.
[47] N. Heidari, P. Moradi, and A. Koochari, "An attention-based deep learning method for solving the cold-start and sparsity issues of recommender systems," *Knowledge-Based Systems*, vol. 256, p. 109835, 2022.
[48] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *Interspeech*, 2017.
[49] K. Song, W. Gao, L. Zhao, J. Lin, C. Sun, and X. Liu, "Cold-start aware deep memory network for multi-entity aspect-based sentiment analysis," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 7 2019, pp. 5197–5203.
[50] B. Hu, Z. Li, and W. Chao, "Data sparsity: A key disadvantage of user-based collaborative filtering?" in *Web Technologies and Applications*, Berlin, Heidelberg, 2012, pp. 602–609.
[51] M. Grčar, D. Mladenič, B. Fortuna, and M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *Advances in Web Mining and Web Usage Analysis*, Berlin, Heidelberg, 2006, pp. 58–76.
[52] F. Yin, Z. Wang, W. Tan, and W. Xiao, "Sparsity-tolerated algorithm with missing value recovering in user-based collaborative filtering recommendation," *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, vol. 10, no. 15, pp. 4939–4948, 2013.
[53] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbaty, "scikit-optimize/scikit-optimize (v0.9.0)," 2021, zenodo. https://doi.org/10.5281/zenodo.5565057.
[54] N. Hug, "Surprise: A python library for recommender systems," *Journal of Open Source Software*, vol. 5, no. 52, p. 2174, 2020.
[55] R. Garnett, *gaussian processes*, 2023, p. 15–44.
[56] S. Rana, C. Li, S. Gupta, V. Nguyen, and S. Venkatesh, "High dimensional bayesian optimization with elastic gaussian process," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, 2017, p. 2883–2891.
[57] "Collaborative filtering," https://spark.apache.org/docs/latest/ml-collaborative-filtering.html, accessed: 2025-01-28.
[58] H. Ha, S. Rana, S. Gupta, T. Nguyen, H. Tran-The, and S. Venkatesh, *Bayesian optimization with unknown search space*, Red Hook, NY, USA, 2019.
[59] V. Nguyen, S. Gupta, S. Rane, C. Li, and S. Venkatesh, "Bayesian optimization in weakly specified search space," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 347–356.
[60] M. Blum and M. A. Riedmiller, "Optimization of gaussian process hyperparameters using rprop," in *The European Symposium on Artificial Neural Networks*, 2013.
[61] R. Preuss and U. von Toussaint, "Parallel computation of Gaussian processes," *AIP Conference Proceedings*, vol. 1853, no. 1, p. 050004, 06 2017.
[62] N. T. Blog, "Netflix recommendations: Beyond the 5 stars," Apr 2012, accessed: 2024-Sep-15.
[63] B. Hallinan and T. Striphas, "Recommended for you: The netflix prize and the production of algorithmic culture," *New Media & Society*, vol. 18, no. 1, pp. 117–137, 2016.
[64] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin, "Do metrics make recommender algorithms?" in *2009 International Conference on Advanced Information Networking and Applications Workshops*, 2009, pp. 648–653.

Tuan-Anh Nguyen received his PhD in computer science from the University of Kent, UK, in 2010. His main research interest is in applying science to broader social activities. He currently works at Ho Chi Minh City University of Foreign Languages - Information Technology.
Email: anhnt1@huflit.edu.vn

Efficient Mining of Concise Patterns for Frequent High-Utility Occupancy Itemsets Using Extended Pruning Strategies

Tien Hoang[1,2,3], Lan Huynh[4], Hai Duong[*,3], Tin Truong[3]

[1] Department of Computer Science, Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] Department of Mathematics and Computer Science, Dalat University, Dalat, Vietnam
[4] Faculty of Information Technology, HCM University of Industry and Trade, Vietnam

Correspondence: Hai Duong. Email: haidv@dlu.edu.vn
Communication: received 01/03/2025, revised 14/04/2025, accepted 20/05/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1360

***Abstract*: Concise representations of frequent high utility occupancy itemsets (FHUOIs), including maximal FHUOIs, closed FHUOIs, and generators of FHUOIs, are essential in utility-driven pattern mining. These representations offer several advantages over the complete set of FHUOIs, such as reduced size, improved efficiency, lower storage costs, and easier analysis. Notably, maximal FHUOIs allow for the recovery of all FHUOIs, while closed FHUOIs and generators enable the generation of non-redundant high utility occupancy rules and the efficient reconstruction of all FHUOIs along with their key information. Despite their significance, existing methods either mine closed FHUOIs and generators separately or lack a solution for mining all maximal FHUOIs. To bridge this gap, this paper introduces two novel algorithms *MaxFHUOI-Miner* and *CGFHUOI-Miner*. The former efficiently extracts only maximal FHUOIs using extended pruning strategies that eliminate non-maximal itemsets early, while the latter simultaneously mines closed FHUOIs and generators by employing innovative pruning techniques to eliminate non-closed and non-generator itemsets at three levels of the prefix tree without performing subset checks. Extensive experiments on real-world and synthetic datasets demonstrate that the proposed algorithms outperform existing and baseline methods in both speed and memory efficiency, particularly for low minimum support and utility occupancy thresholds in dense databases.**

Keywords: *Frequent high utility occupancy itemset; Pruning strategy; Weak upper bound; Weak lower bound, Concise representations*

I. INTRODUCTION

Frequent High Utility Occupancy Itemset (FHUOI) mining (FHUOIM) is a fundamental task in data mining that focuses on identifying valuable patterns in quantitative transaction databases (QTDBs). It aims to discover itemsets that not only occur frequently but also contribute significantly to the total utility of transactions [1][2][3][4]. In FHUOIM, the utility occupancy of an itemset A in a transaction T measures its contribution to T 's overall utility. The total utility occupancy of A is calculated as the average occupancy across all transactions in which it appears. An itemset qualifies as an FHUOI if both its average utility occupancy and support exceed predefined thresholds. FHUOIM is designed to efficiently extract all such FHUOIs from a QTDB. This technique has various real-world applications, such as improving recommendation systems based on mobile application traffic patterns and analyzing web log data for user behavior insights [5][6].

A main challenge in FHUOIM is the large number of patterns generated, especially in dense QTDBs or when the support and utility occupancy thresholds are low. This leads to longer execution times, higher memory usage, and makes it difficult for users to interpret and analyze the results. A more effective solution is to focus on concise representations of FHUOIs, such as closed FHUOIs (CFHUOIs) [7], maximal FHUOIs (MFHUOIs), and generators of FHUOIs (GFHUOIs) [8][9], rather than extracting all FHUOIs. These three representations help reduce redundancy and improve efficiency. An MFHUOI is an FHUOI that is not strictly contained in any other FHUOI. The $\mathcal{MFHUOI}$ set of all MFHUOIs is usually much smaller than the full set $\mathcal{FHUOI}$ of all FHUOIs, making it more manageable for users. Additionally, MFHUOIs can be used to reconstruct the complete set of FHUOIs. A CFHUOI is an FHUOI that has no larger FHUOI with the same support, while a GFHUOI is an FHUOI with no smaller FHUOI having the same support. The $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets of CFHUOIs and GFHUOIs are generally smaller than

$\mathcal{FHUOI}$ but larger than the $\mathcal{MFHUOI}$ set. Notably, CFHUOIs and GFHUOIs represent the largest and smallest elements, respectively, within each equivalence class. In this context, an equivalence class is defined as a group of itemsets that share identical support and occur in the same set of transactions in the database [10].

In this paper, we focus on mining the $\mathcal{MFHUOI}$, $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets for several key reasons. First, these sets share a major advantage: they are significantly smaller than the $\mathcal{FHUOI}$ set of all FHUOIs. This allows for more efficient discovery, reduced storage requirements, and easier interpretation for users. Second, each set offers distinct benefits. The $\mathcal{MFHUOI}$ set is the smallest among the three and can be used to reconstruct all FHUOIs. Additionally, MFHUOIs are valuable for identifying frequent longest common patterns in real-world applications. The $\mathcal{CFHUOI}$ set, while larger than $\mathcal{MFHUOI}$, is lossless, meaning it retains complete frequency information for all itemsets. CFHUOIs are useful because they represent the largest itemsets with high utility occupancy in specific transaction groups (such as customer groups). They are the most representative patterns in their equivalence classes, itemsets that appear in the same transactions. This makes CFHUOIs highly applicable in practical scenarios. For the $\mathcal{GFHUOI}$ set, researchers argue that it aligns with the minimum description length principle [10], making it preferable because GFHUOIs are the smallest representatives of their equivalence classes. Finally, CFHUOIs and GFHUOIs can be combined to form non-redundant high utility occupancy association rules, where a GFHUOI appears on the left-hand side and a CFHUOI on the right-hand side. They also enable fast reconstruction of all FHUOIs in equivalence classes, along with their support values.

The above advantages emphasize the importance of mining three sets $\mathcal{MFHUOI}$, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$ and drive the need for optimizing existing algorithms to extract these sets more efficiently.

Contributions. As discussed earlier, each of the three condensed representations of FHUOIs serves distinct purposes depending on the mining objective. In many scenarios, the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets are extracted together to generate non-redundant high utility occupancy rules and efficiently recover all FHUOIs, along with their supports and utility occupancies. In contrast, $\mathcal{MFHUOI}$, being the smallest set, is often mined separately for specific applications. Despite their importance, to our knowledge, no existing algorithm can simultaneously mine both $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$, or independently extract $\mathcal{MFHUOI}$. To bridge this gap, this paper introduces two novel algorithms: one for concurrently discovering both $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$, and another dedicated to efficiently mining $\mathcal{MFHUOI}$ alone.

The main contributions of this paper are as follows. First, we introduce two extended pruning strategies, EPNonMaxBr and LPNonMaxBr, designed to eliminate unpromising branches early in the prefix tree that cannot contain any MFHUOIs. These strategies are integrated into a novel algorithm, MaxFHUOI-Miner, to efficiently extract the $\mathcal{MFHUOI}$ set. Second, we propose two pruning techniques, EPNonCGBr and LPNonCGBr, that reduce the search space by eliminating branches that cannot contain both CFHUOIs and GFHUOIs. Using these techniques, we develop CGFHUOI-Miner, a novel algorithm that discovers both CFHUOIs and GFHUOIs within a single process. To the best of our knowledge, CGFHUOI-Miner and MaxFHUOI-Miner are the first algorithms specifically designed for mining CFHUOIs and GFHUOIs together and MFHUOIs independently, respectively. Finally, we provide experimental evidence demonstrating that the proposed algorithms outperform existing and baseline methods in terms of runtime, memory efficiency, and scalability.

The rest of this paper is structured as follows. Section 2 reviews related work, followed by Section 3, which introduces fundamental concepts and notation for identifying FHUOIs and their concise representations. Section 4 details the core theoretical contributions, including innovative pruning strategies to efficiently filter out non-closed, non-generator, and non-maximal FHUOIs, along with the introduction of two novel algorithms, MaxFHUOI-Miner and CGFHUOI-Miner. Section 5 presents the experimental analysis of these algorithms. Finally, Section 6 summarizes the findings and discusses potential directions for future research.

II. RELATED WORK

1. Mining high utility itemsets and high utility occupancy itemsets

The problem of high utility itemset (HUI) mining (HUIM) in QTDBs [11] has been a major focus of research in recent years. A significant challenge in HUIM is that the utility function u does not exhibit the anti-monotonic (or Downward Closure) property, which is a key principle for efficiently reducing the search space in pattern mining. To tackle this limitation, researchers have introduced upper bounds on u, including the anti-monotonic Transaction Weighted Utility (TWU) [12] and a more refined upper bound that accounts for the remaining utility of a pattern in its supporting transactions. This enhanced bound is implemented in the HUI-Miner algorithm, which is built on a utility-list-based framework [13]. To address the

HUIM problem, various algorithms have been developed. UP-Growth [14] employs a utility pattern tree (UP-Tree) while requiring only two scans of the database. EFIM [15] enhances mining efficiency by incorporating two upper bounds: sub-tree utility and local utility. HMiner [16] utilizes a compact utility list along with a virtual hyperlink data structure to efficiently manage itemset information. Meanwhile, Hamm [17] is a single-phase algorithm that introduces a novel data structure, TV, which combines a prefix tree with a utility vector for improved performance.

High Utility Itemset Mining (HUIM) has been widely applied in various domains; however, the conventional utility measure u of an itemset A in a supporting transaction T (also referred to as the absolute utility of A in T [18]) does not adequately capture its utility occupancy level within T. Utility occupancy quantifies the relative contribution of A to the total transaction utility, providing a more granular assessment of its significance for each transaction. This improved measure is key to accurately finding important patterns in real-world applications, like personalized app and webpage recommendations [5]. To address this limitation, Shen et al. [5] introduced the concept of utility occupancy in QTDBs to extend occupancy-based analysis. The utility occupancy of an itemset A in a supporting transaction T is defined as the ratio of its utility in T to the total utility of T, representing the relative utility of A in T. This measure highlights the importance of A in terms of utility distribution across transactions. The overall utility occupancy of A in a QTDB, denoted as $occ(A)$, is computed as the average of its utility occupancies across all transactions containing A. An itemset is considered a Frequent High Utility Occupancy Itemset (FHUOI) if its support and utility occupancy meet or exceed the user-defined min_sup and min_uo thresholds, respectively. This constraint ensures that extracted patterns are not only frequent but also highly relevant in terms of utility contribution, making them valuable for various analytical and decision-making tasks in data mining.

The problem of Frequent High Utility Occupancy Itemset Mining (FHUOIM) aims to identify all FHUOIs in a QTDB. This task has been widely applied in various domains, particularly in recommendation systems. For example, in mobile app traffic analysis, each mobile application is treated as an item, and its data usage represents its utility. By mining FHUOIs, analysts can address critical questions such as: "Which essential mobile applications contribute the most to users' overall data consumption?" [5]. Traditional pattern mining methods, which focus solely on high utility or high frequency, are insufficient for answering such queries, as they do not account for the relative importance of an item's utility in each transaction. Another key application is webpage recommendation based on web browsing logs, where each website is considered as an item, and its browsing duration serves as its utility. Identifying FHUOIs provides deeper insights into user engagement, enabling more tailored content recommendations and enhanced user experiences. By capturing both utility and occupancy, FHUOIM provides a more comprehensive and informative approach to pattern discovery in transactional datasets.

A key challenge in FHUOIM is that the occupancy measure (occ), like utility, lacks anti-monotonicity, making it difficult to efficiently prune the search space. To tackle this problem, the authors in [5] introduced an upper bound (UB) on occ and proposed the OCEAN algorithm to facilitate FHUOIM by reducing unnecessary computations. To further enhance the efficiency of FHUOIM algorithms, Gan et al. [19] developed the HUOPM algorithm, which efficiently discovers the complete set of FHUOIs. This algorithm efficiently removes infrequent or low utility occupancy itemsets by using the anti-monotonicity of support and an upper bound on occ. More recently, He et al. [6] introduced SHO, a method specifically designed for handling large-scale datasets. By employing a suffix-based partitioning technique, SHO significantly improves computational efficiency, allowing for faster and more scalable FHUOIM. Beyond these developments, researchers have expanded FHUOIM to new problem settings. For instance, some studies have focused on mining potential FHUOIs in uncertain QTDBs [20], while others have explored frequent high utility occupancy sequential pattern mining in quantitative sequence databases [21]. These extensions broaden the applicability of FHUOIM, making it more suitable for uncertain and sequential data environments in real-world applications.

2. Mining concise representations for HUIs and FHUOIs

High utility itemsets (HUIs) are important in many real-world applications, but their cardinality can be excessively large, especially in large, dense QTDBs or when the minimum utility threshold is too low. This makes analyzing and utilizing HUIs both difficult and time-consuming. To address this challenge, researchers have developed algorithms that generate compact representations of HUIs, significantly reducing the result set while retaining essential information. Notable methods for this include CHUI-Miner(Max) [22] and MaxC-HUIM [23] for mining maximal HUIs; CHUD [18], CHUI-Miner [24], EFIM-Closed [25], HMiner-Closed [26], and C-HUIM [23] for exploiting closed HUIs; and approaches introduced in [27] for high utility generators. A high utility pattern A is maximal if no proper super-

itemset of A is an HUI. It is closed (or a generator) if no proper HUI super-itemset (or sub-itemset, respectively) of A has the same support. Note that maximal HUIs provide a compact representation of HUIs, with their cardinality being smaller than those of closed HUIs and HUI generators. In contrast, closed and generator HUIs not only offer concise representations but also retain all essential information, ensuring a lossless representation of HUIs [18]. Notably, within each equivalence class of HUIs that share the same supporting transactions, there exists a unique closed HUI, which represents the largest pattern in the class under the traditional set inclusion relation ($\subseteq$). However, multiple HUI generators may exist in the same class, each corresponding to a minimal pattern [18][27].

The challenge of managing FHUOIs is similar, as their cardinality often becomes excessively large, making both analysis and practical application difficult. This underscores the necessity of discovering concise representations for FHUOIs. To address this issue, the CloFHUOIM algorithm [7] was introduced as the first efficient method for extracting compact representations of FHUOIs by identifying the $\mathcal{CFHUOI}$ set, which consists of all closed FHUOIs (CFHUOIs). Furthermore, the authors in [7] proposed the MaxCloFHUOIM algorithm, which simultaneously discovers both CFHUOIs and maximal FHUOIs (MFHUOIs) in a single process. To enhance efficiency, a novel weak upper bound, $wubocc$, was introduced for the utility occupancy function. This bound is tighter than the previous upper bound $\widehat{\Phi}$ [19] and is used in an effective pruning strategy to reduce the search space. Additionally, two algorithms have been developed for mining the $\mathcal{GFHUOI}$ set of all generators of FHUOIs (GFHUOIs): GFHUOI-Miner and Gen-FHUOIM [8][9]. Since the occ function lacks the monotonicity property in each equivalence class, the authors devised weak lower bounds on occ and corresponding pruning techniques to efficiently eliminate non-generator FHUOIs early in the mining process. Experimental results confirm that these algorithms significantly outperform baseline approaches in terms of runtime, memory consumption, and scalability on large datasets.

III. PRELIMINARIES AND PROBLEM DEFINITION

This section presents concepts and notation related to the task of mining frequent high utility occupancy itemsets (FHUOIs) and their concise representations in QTDBs. The input data format for this task is formally defined as follows.

Definition 1 *(Quantitative transaction database).* Let $\mathcal{A} \overset{\text{def}}{=} \{a_j, j \in J \overset{\text{def}}{=} \{1, 2, ..., N'\}\}$ represent a finite set of distinct items, and let $X \subseteq \mathcal{A}$ denote a subset of $\mathcal{A}$. If X contains exactly k items, it is referred to as a k-itemset, where $k = |X|$. Without loss of generality, we assume that the items in each itemset are ordered according to a total order $\prec$ (e.g., lexicographic order). The profit vector $\mathcal{P} \overset{\text{def}}{=} \left(p\left(a_j\right), j \in J \text{ and } p\left(a_j\right) > 0\right)$ represents the external utility of all items in the set $\mathcal{A}$, where $p\left(a_j\right)$ denotes the relative importance of each item a_j. This importance can be quantified by attributes such as unit profit, price, weight, or other relevant factors. A q-item is defined as a pair (a, q), where a is an item from $\mathcal{A}$ and q is a positive integer representing the purchase quantity or internal utility of item a. A quantitative database (QTDB) $\mathcal{D}$ is defined as a collection of transactions, $\mathcal{D} \overset{\text{def}}{=} \left\{T_i, i \in I \overset{\text{def}}{=} \{1, 2, \ldots, N\}\right\}$, where each transaction T_i is represented as $T_i \overset{\text{def}}{=} \left\{\left(a_j, q_{ij}\right), j \in J_i\right\}$, consisting of a set of q-items. Each transaction T_i is uniquely identified by a transaction identifier (TID), with TID = i and J_i is a subset of J, $J_i \subseteq J$. The utility of a q-item (a, q), denoted as $u((a, q))$, is defined as:

$$u((a, q)) \overset{\text{def}}{=} q \times p(a) \tag{1}$$

where $p(a)$ is the external utility of item a. Intuitively, this utility represents the profit generated by the sale of item a within the transaction where the q-item (a, q) apprears.

Definition 2 *(Integrated QTDB).* To eliminate redundant computations of the utility u for q-items $\left(a_j, q_{ij}\right)$ across transactions T_i in $\mathcal{D}$, these utility values are precomputed and stored in an integrated quantitative database, referred to as $\mathcal{D}'$, which is defined as:

$$\mathcal{D}' \overset{\text{def}}{=} \left\{T_i' \overset{\text{def}}{=} \left\{\left(a_j, q_{ij}'\right), j \in J_i\right\}, J_i \subseteq J, i \in I\right\} \tag{2}$$

where $q_{ij}' = q_{ij} * p\left(a_j\right)$.

Table I
A QTDB $\mathcal{D}$

TID	Transaction (item, quantity)
T_1	$(a, 1), (e, 8)$
T_2	$(b, 1), (c, 12), (d, 12), (e, 21), (f, 1)$
T_3	$(a, 2)$
T_4	$(a, 1), (b, 5), (c, 2), (d, 4), (e, 26)$
T_5	$(a, 3), (e, 9), (f, 1)$

Table II
A PROFIT VECTOR $\mathcal{P}$

a_j	a	b	c	d	e	f
$p\left(a_j\right)$	16	2	1	3	2	55

Running Example: Table 1 and Table 2 present a QTDB $\mathcal{D}$ and an external utility table $\mathcal{P}$, respectively. Table 3 illustrates the integrated QTDB $\mathcal{D}'$, obtained by combining $\mathcal{D}$ and $\mathcal{P}$. This integrated database, $\mathcal{D}'$, serves as a running example throughout the study to illustrate key concepts.

Table III
THE INTEGRATED QTDB $\mathcal{D}'$

Tid	Quantitative Transaction	$tu(T_i)$
T_1	$(a,16),(e,16)$	32
T_2	$(b,2),(c,12),(d,36),(e,42),(f,55)$	147
T_3	$(a,32)$	32
T_4	$(a,16),(b,10),(c,2),(d,12),(e,52)$	92
T_5	$(a,48),(e,18),(f,55)$	121
	$TU=$	424

For clarity and brevity, each transaction T_i' in $\mathcal{D}'$ will be referred to as T_i, and the shorthand ae will represent the itemset $\{a,e\}$.

Definition 3 *(Support of an Itemset).* For any $X \subseteq \mathcal{A}$, where $X \neq \varnothing$, let $\rho(X)$ represent the set of transactions in $\mathcal{D}'$ that contain X.

$$\rho(X) \stackrel{\text{def}}{=} \left\{T_i \in \mathcal{D}' \mid q'_{ij} > 0, \forall a_j \in X\right\} \tag{3}$$

By convention, $\rho(\varnothing) \stackrel{\text{def}}{=} \mathcal{D}'$. The support of X, denoted as $supp(X)$, is defined as the number of transactions in the QTDB $\mathcal{D}'$ that contain X. Formally, $supp(X)$ is the cardinality of the set $\rho(X)$. Only non-empty itemsets X with at least one occurrence in the transactions of $\mathcal{D}'$ are considered, i.e., $X \in \mathcal{IS} \stackrel{\text{def}}{=} \{X \subseteq \mathcal{A} \mid X \neq \emptyset \wedge \rho(X) \neq \varnothing\}$.

Example 1. For itemset $X = ae$, as $X \subseteq T_1, X \subseteq T_4$ and $X \subseteq T_5$, we have $\rho(X) = \{T_1, T_4, T_5\}$ and $supp(X) = 3$.

Definition 4 *(Itemset utility in a transaction and utility function).* The utility of an item a_j (where $j \in J$) in a transaction $T_i \in \mathcal{D}'$ is denoted and defined as $u\left(a_j, T_i\right) \stackrel{\text{def}}{=} q'_{ij}$. For an itemset X in T_i, where T_i contains X, the utility of X in T_i is denoted as $u(X,T_i) \stackrel{\text{def}}{=} \sum_{a_j \in X} u\left(a_j, T_i\right)$. The utility function u of an itemset X across $\mathcal{D}'$ is denoted by $u(X)$ and defined as: $u(X)$ and $u(X) \in R_+ \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid x > 0\}$.

$$u(X) \stackrel{\text{def}}{=} \sum_{T_i \in \rho(X)} u(X,T_i) \tag{4}$$

where $u(X,T_i) \stackrel{\text{def}}{=} \sum_{a_j \in X} u\left(a_j, T_i\right)$

Definition 5 *(Transaction utility and total utility).* The utility of a transaction T_i is denoted $tu(T_i)$ and is defined by $tu(T_i) \stackrel{\text{def}}{=} \sum_{\left(a_j, q'_{ij}\right) \in T_i} q'_{ij}$. The total utility (TU) of $\mathcal{D}'$ is the sum of the utilities of all transactions in $\mathcal{D}'$, expressed as

$$TU \stackrel{\text{def}}{=} \sum_{T_i \in \mathcal{D}} tu(T_i) \tag{5}$$

where $tu(T_i) \stackrel{\text{def}}{=} \sum_{\left(a_j, q'_{ij}\right) \in T_i} q'_{ij}$.

Example 2. For itemset $X = ae$, then $\rho(X) = \{T_1, T_4, T_5\}, u(X,T_1) = 16+16 = 32$, $u(X,T_4) = 16+52 = 68$, and $u(X,T_5) = 48+18 = 66$. Then, $u(X) = u(X,T_1) + u(X,T_4) + u(X,T_5) = 32+68+66 = 166$. We have $tu(T_1) = 16+16 = 32$ and $TU = 32+147+32+92+121 = 424$.

Definition 6 *(Utility Occupancy).* The utility occupancy of an itemset $X \in \mathcal{A}$ in $\mathcal{D}'$ is denoted as $occ(X)$ and defined as follows:

$$occ(X) \stackrel{\text{def}}{=} \frac{\sum_{T_i \in \rho(X)} u_{rel}(X,T_i)}{supp(X)}, \tag{6}$$

where $u_{rel}(X,T_i) \stackrel{\text{def}}{=} \frac{u(X,T_i)}{tu(T_i)}$ represents the utility occupancy ratio of X relative to the utility of the input transaction T_i.

Definition 7 *(Frequent High Utility Occupancy Itemset).* Given two user-defined positive thresholds, minimum utility occupancy $min\_uo \in (0,1]$ and minimum support min_sup, an itemset X is called a high utility occupancy itemset (HUOI) if its utility occupancy in $\mathcal{D}'$ satisfies the condition $occ(X) \geq min\_uo$. Conversely, if $occ(X) < min\_uo$, then X is termed a low utility occupancy itemset (LUOI). Additionally, an HUOI X is further classified as a frequent high utility occupancy itemset (FHUOI) if its support meets or exceeds the threshold min_sup, i.e., $supp(X) \geq min\_sup$.

The problem of FHUOI mining ($\mathcal{FHUOIM}$) is to identify the set of all FHUOIs, defined as

$$\mathcal{FHUOI} \stackrel{\text{def}}{=} \{X \in \mathcal{A} \mid occ(X) \geq min\_uo \wedge supp(X) \geq min\_sup\} \tag{7}$$

Equivalently, this set can be expressed as $\mathcal{FHUOI} = \mathcal{HUOI} \cap \mathcal{FI}$, where $\mathcal{HUOI} \stackrel{\text{def}}{=} \{X \in \mathcal{A} \mid occ(X) \geq min\_uo\}$ and $\mathcal{FI} \stackrel{\text{def}}{=} \{X \in \mathcal{A} \mid supp(X) \geq min\_sup\}$ are the sets of all HUOIs and frequent itemsets (FIs), respectively. It is important to note that $\mathcal{FHUOIM}$ generalizes the traditional HUOI mining problem, where $min\_sup = 1$ represents a special case.

Example 3. For $min\_sup = 2$ and $min\_uo = 0.7$, consider the itemset $X = ae$. Then, $\rho(X) = \{T_1, T_4, T_5\}, supp(X) = 3$, we have $u_{rel}(X,T_1) = (16+16)/32 = 1$, $u_{rel}(X,T_4) = (16+52)/92 = 0.739$ and $u_{rel}(X,T_5) = (48+18)/121 = 0.545$. The utility occupancy of X in $\mathcal{D}'$ is $occ(X) = (1+0.739+0.545)/3 = 0.762 \geq min\_uo$ and $supp(X) \geq min\_sup$. Thus, X is an FHUOI. Meanwhile, the itemset $Y = c$ is a low utility occupancy itemset because $occ(Y) = 0.0517 < min\_uo$.

We assume that all items in $\mathcal{A}$ are ordered according to a total order relation $\prec$. For any item y and itemsets X and Y, we denote $X \prec y$ if and only if (or briefly, iff) $x \prec y, \forall x \in X$, or $X \prec Y$ iff $x \prec y, \forall x \in X, \forall y \in Y$. If $X \cap Y = \emptyset$, the union of the two disjoint itemsets X and Y is denoted as $X + Y$. Furthermore, if $X \cap Y = \emptyset$ and $X \prec Y$, the notation X + Y is replaced by $X \oplus Y$ to indicate the ordered union of the two disjoint itemsets.

Definition 8 *(Forward and Backward Extensions of an Itemset)*. For any two itemsets X and Y where $X \cap Y = \emptyset$ and $X \prec Y$, the itemset $S = X \oplus Y$ is referred to as a forward extension (FE), or simply an extension, of X with Y. If $Y \neq \emptyset$, then S is called a proper FE of X. Given an itemset $Y = X \oplus y$ with $X \prec y$, a new itemset $B = (X + P) \oplus y$, where $P \prec y$, is formed by inserting the items of P into X before y, in accordance with the order relation $\prec$. This itemset B is referred to as a backward extension of Y.

Specifically, given an itemset $Y = X \oplus y$, where Y is formed by appending an item y to X, if $supp(Y) = supp(X)$, then Y is defined as a $1-$ forward of X (with y). Similarly, if $supp(C) = supp(Y)$ for $Y = X \oplus y \subset C = (X + x) \oplus y$ with $x \notin X, X \prec y$ and $x \prec y$, meaning C is derived from Y by inserting an item x before the last item y of Y, then C is referred to as a $1-$ backward of Y (with x). It is evident that a $1-$ forward of X also constitutes a proper forward extension of X.

The operator $\oplus$ plays a crucial role in high utility occupancy itemset mining (HUOIM) algorithms, as it enables the systematic exploration of larger itemsets in the search space.

Example 4. For instance, consider $Y = bd$. Then, $bde = Y \oplus e$ is a $1-$ forward of Y with e because $supp(Y) = supp(bde) = 3$. However, bdf is not a $1-$ forward of bd since $supp(bdf) = 1 \neq supp(bd) = 2$, although bdf is an FE of bd. Similarly, the itemset bcd is also a $1-backward$ of Y (with c) as $supp(Y) = supp(bcd) = 2$. In contrast, abd is not a $1-backward$ of Y(with a) since $supp(Y) = 2 \neq supp(abd) = 1$.

Definition 9 (*Set $\mathcal{CFHUOI}$*). Let X be an FHUOI. The itemset X is termed a closed frequent high utility occupancy itemset (CFHUOI) if no FHUOI exists as a proper superitemset of X with the same support. The set of all CFHUOIs is denoted and defined as:

$$\mathcal{CFHUOI} \stackrel{\text{def}}{=} \{X \in \mathcal{FHUOI} \mid \nexists Y \in \mathcal{FHUOI} : Y \supset X \wedge supp(Y) = supp(X)\} \tag{8}$$

Definition 10 (*Set $\mathcal{GFHUOI}$*). Let X be an FHUOI, i.e., $X \in \mathcal{FHUOI}$. X is referred to as a generator of frequent high utility occupancy itemset (GFHUOI) if no FHUOI exists as a proper sub-itemset of X with the same support. The collection of all such GFHUOIs is denoted as $\mathcal{GFHUOI}$, which is formally defined as:

$$\mathcal{GFHUOI} \stackrel{\text{def}}{=} \{X \in \mathcal{FHUOI} \mid \nexists Y \in \mathcal{FHUOI} : Y \subset X \wedge supp(Y) = supp(X)\} \tag{9}$$

Definition 11 (*Set $\mathcal{MFHUOI}$*). An FHUOI X is said to be *a maximal frequent high utility occupancy itemset* (MFHUOI) if *no FHUOI exists as a proper* superset of X. The set of all MFHUOIs is denoted and defined as

$$\mathcal{MFHUOI} \stackrel{\text{def}}{=} \{X \in \mathcal{FHUOI} \mid \nexists Y \in \mathcal{FHUOI} : Y \supset X\} \tag{10}$$

Problem statement. Given a quantitative transaction database (QTDB) $\mathcal{D}'$ and two userdefined thresholds ($min\_uo \in (0; 1]$ for minimum utility occupancy and min_sup for minimum support), this study develops two efficient algorithms: (1) for simultaneously mining both $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets, and (2) for discovering the $\mathcal{MFHUOI}$ set separately.

Example 5. Consider $min\_uo = 0.4$, $min\_sup = 2$, and $X = ef$. We have $occ(X) = 0.63 \geq min\_uo$ and $supp(X) = 2 \geq min\_sup$. Since no FHUOI exists as a superset of X with the same support, X qualifies as a CFHUOI. However, X is not a GFHUOI, because there exists an FHUOI $Y = f \in \mathcal{FHUOI}$ with $occ(Y) = 0.41$, $X \supset Y$, and $supp(X) = supp(Y) = 2$. Y is classified as a GFHUOI since $\nexists Z \in \mathcal{FHUOI} : Y \supset Z$ and $supp(Z) = supp(Y)$. In addition, $Z = bcde$ is identified as an MFHUOI as $occ(Z) = 0.73, supp(Z) = 2$, and $\nexists Y \in \mathcal{FHUOI} : Y \supset Z$. Similarly, we have $\mathcal{GFHUOI} = \{a, ae, f, bd, be, ce, de\}$, $\mathcal{CFHUOI} = \{a, ae, ef, bcde\}$ and $\mathcal{MFHUOI} = \{ae, ef, bcde\}$. For $min\_uo = 0.1$ and $min\_sup = 1$, then $|\mathcal{GFHUOI}| = 15$, $|\mathcal{CFHUOI}| = 8$, and $|\mathcal{MFHUOI}| = 3$, while $|\mathcal{FHUOI}| = 47$.

For clarity, Table 4 summarizes the key abbreviations and notations used throughout this paper.

Table IV
ABBREVIATIONS AND NOTATIONS

Abbreviation	Explanation
HUOI	A high utility occupancy itemset
FI	A frequent itemset A, supp$(A) \geq min_sup$
FHUOI	A frequent high utility occupancy itemset ($occ(A) \geq min\_uo$ and $supp(A) \geq min\_sup$)
$\mathcal{HUOI}$	The set of all HUOIs
$\mathcal{FI}$	The set of all FIs
$\mathcal{FHUOI}$	The set of all FHUOIs, $\mathcal{FHUOI} = \mathcal{HUOI} \cap \mathcal{FI}$
CFHUOI	A closed frequent high utility occupancy itemset
GFHUOI	A generator of frequent high utility occupancy itemset
MFHUOI	A maximal frequent high utility occupancy itemset
$\mathcal{CFHUOI}$	The set of all CFHUOIs
$\mathcal{GFHUOI}$	The set of all GFHUOIs
$\mathcal{MFHUOI}$	The set of all MFHUOIs
$wubocc$	A weak upper bound (WUB) on occ
$wlbocc$	A novel weak lower bound (WLB) on occ
$NonC\_FHUO$	A non-closed frequent high utility occupancy
$NonG\_FHUO$	A non-generator frequent high utility occupancy
$NonCG\_FHUO$	A non-closed and non-generator frequent high utility occupancy

Consider the process of searching for candidate itemsets to identify the set of $\mathcal{FHUOI}$ or its concise representa-

tions, such as $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$, performed on a prefix search tree (or briefly, prefix tree). In this context, a prefix tree is a data structure that helps organize itemsets by grouping those with the same prefixes [28]. The tree starts from an empty itemset at the root, and each node represents one item. The path from the root to any node forms an itemset. We explore the tree using depth-first search (DFS), following a predefined item order $\prec$. For simplicity, in this paper, each node P stands for a complete itemset, and each child of P represents one of its forward extensions (see Figure 1). We denote the branch $Br(X)$ as the set containing X and all its proper FEs, and the proper branch $PBr(X)$ as $Br(X)$ excluding the itemset X itself.

IV. MINING CONCISE REPRESENTATIONS OF FHUOIS

Note that the goal of this paper is to discover the $\mathcal{CFHUOI}, \mathcal{GFHUOI}$ and $\mathcal{MFHUOI}$ sets. To enhance mining efficiency, it is crucial to prune unpromising branches $Br(A)$ or $PBr(A)$) from the prefix tree rooted at itemset A as early as possible. They include: (1) infrequent itemsets, (2) low utility occupancy itemsets (LUOIs), and (3) non-closed, non-generator, or non-maximal FHUOIs, i.e., FHUOIs that are neither closed, generators, nor maximal patterns. Infrequent itemsets can be eliminated early using the anti-monotonicity property of support: if an itemset A is infrequent, all its supersets are also infrequent. Thus, when A is infrequent, its entire branch $Br(A)$ can be pruned immediately. However, frequent itemsets can still be numerous, especially in large QTDBs with low min_sup values. To address this, the next sections introduce efficient pruning strategies to quickly discard LUOIs and non-closed, non-generator, or non-maximal FHUOIs.

1. Pruning low utility occupancy itemsets

Since the function occ is not anti-monotonic ($\mathcal{AM}$), the mining process requires upper bounds (UBs) or weak UBs (WUBs) on occ that exhibit $\mathcal{AM}$ or $\mathcal{AM}$-like properties [29] to facilitate the early elimination of LUOIs. A function ub is a UB on occ if it satisfies $occ(C) \leq ub(C), \forall C \in \mathcal{IS}$. Similarly, a function wub is a WUB on occ if $occ(S) \leq wub(C)$ for any itemset C and its proper FE S, where $S = C \oplus D \supset C$ with $D \neq \emptyset$. Among two UBs, ub_1 is considered tighter than ub_2 if $ub_1(C) \leq ub_2(C), \forall C \in \mathcal{IS}$.

Two existing (W)UBs on occ have been introduced for FHUOIM: the UB $\widehat{\Phi}$ [19] and the WUB $wubocc$ [7]. Since $wubocc$ is proven to be tighter than $\widehat{\Phi}$, this study employs $wubocc$ for early LUOI elimination based on the EPLUOI pruning strategy [7]. The definition and properties of $wubocc$ are presented below.

Let us denote $h(A) \stackrel{\text{def}}{=} \{T_i \in \mathcal{D}' \mid T_i \supseteq A \oplus E, E \neq \emptyset\}$ as the set of all transactions that contain at least a proper FE of A, so $h(A)$ is a subset of $\rho(A)$.

Definition 12 (*WUB wubocc on occ* [7]). For any itemset A and transaction T_i such that $|h(A)| \geq min\_sup$ and $T_i \in h(A)$, let us denote $\text{rem}(A, T_i)$ as an itemset that contains all remaining items in T_i after A. Let define two functions $ub_r(A, T_i) \stackrel{\text{def}}{=} u(A, T_i) + u(\text{rem}(A, T_i))$ and $ub_{r\_rel}(A, T_i) \stackrel{\text{def}}{=} \frac{ub_r(A,T_i)}{tu(T_i)}$. After sorting the series $\{ub_{r\_rel}(A, T_i) \mid T_i \in h(A)\}$ in descending order to create the new series $\{wubocc_i, i = 1..n\}$, the WUB $wubocc$ on occ is defined as follows:

$$wubocc(A) \stackrel{\text{def}}{=} \frac{1}{min\_sup} \sum_{1 \leq i \leq min\_sup} wubocc_i. \quad (11)$$

Proposition 1 (*The property of wubocc* [7]). $wubocc$ is a WUB on occ, i.e. $occ(B) \leq wubocc(A)$ for any itemset $B = A \oplus E$ being the proper FE of A with $E \neq \emptyset$.

Pruning strategy EPLUOI (*Early Pruning of LUOIs*). If $wubocc(A) < min\_uo$, the search for FHUOIs in the proper branch $PBr(A)$ is terminated, as Proposition 1 guarantees that no FHUOI exists in this proper branch.

Example 6. For $min\_sup = 1$ and $min\_uo = 0.93$, consider the itemset $X = bd$. Since $\rho(X) = h(X) = \{T_2, T_4\}$, then $ub_{r\_rel}(X, T_2) = \frac{2+36+42+55}{147} = 0.918$ and $ub_{r\_rel}(X, T_4) = \frac{10+12+52}{92} = 0.804$. After sorting the series $\{0.918, 0.804\}$ in descending order to create the new series $\{wubocc_i, i = 1..2\} = \{0.918, 0.804\}$, we have $wubocc(A) \stackrel{\text{def}}{=} \frac{1}{min\_sup} \sum_{1 \leq i \leq min\_sup} wubocc_i = wubocc_1 = 0.918 < min\_uo$. Thus, $PBr(X)$ is eliminated early.

For brevity, if $C \notin \mathcal{FHUOI}$ or C is *not closed* (or *not maximal*), then C is referred to as *a non-closed* FHUO ($NonC\_FHUO$) (or *non-maximal* FHUO, $NonM\_FHUO$) itemset. Similarly, if $C \notin \mathcal{FHUOI}$ or C is not a generator, then it is called a non-generator FHUO ($NonG\_FHUO$) itemset. If $Br(C)$ cannot include any CFHUOIs (or GFHUOIs), it is also referred to as a $NonC\_FHUO$ (or $NonG\_FHUO$) branch. In addition, if it cannot contain both CFHUOIs and GFHUOIs, it is also called a $NonC\_FHUO$ and $NonG\_FHUO$ branch, or briefly, $NonCG\_FHUO$ branch.

2. Extended pruning strategies of non-closed or non-generator branches

Even after removing infrequent itemsets and LUOIs, a QTDB can still generate numerous FHUOIs. Thus, extracting CFHUOIs and GFHUOIs is challenging, as it requires checking inclusion relationships to filter out non-closed and non-generator FHUOIs. A key challenge is how to efficiently prune $NonC\_FHUO$ and $NonG\_FHUO$ branches

from the search space. Since the occ function is monotonic within each equivalence class, $NonC\_FHUO$ branches can be pruned using strategies from [7]. However, the early elimination of $NonG\_FHUO$ branches requires an anti-monotonic function in each equivalence class, which occ lacks. To address this issue, the authors of [8] introduced the weak lower bound (WLB) $wlbocc$ on occ. For any itemset A, a function $wlb(A)$ is a WLB on $occ(A)$ if $wlb(A) \leq occ(B)$ for any proper FE B of A. The $wlbocc$ WLB is defined as follows.

Definition 13 (*WLB $wlbocc$ on occ* [8]). For any strictly frequent itemset X (i.e. $l \overset{\text{def}}{=} |h(X)| \geq min\_sup$) and transaction $T_i \in h(X)$, let us define $\mathbb{Q}(X,T_i) \overset{\text{def}}{=} u(X,T_i) + \min\left\{u\left(a_j,T_i\right) \mid a_j \in \text{rem}(X,T_i)\right\}$, and $\mathbb{Q}_{rel}(X,T_i) \overset{\text{def}}{=} \mathbb{Q}(X,T_i)/tu(T_i)$. Let $\mathcal{F} = \{\mathbb{Q}_{rel}(X,T_i) \mid T_i \in h(X)\}$ represent the set of relative weights $\mathbb{Q}_{rel}(X,T_i)$ across all transactions in $h(X)$. Assume that the series $\mathcal{F}$ is sorted in ascending order to create the new one $\mathcal{F}' = \{\tau_i, i = 1..l\}$. Then, the $wlbocc$ WLB is defined as follows:

$$wlbocc(X) = \frac{1}{min\_sup} \sum_{1 \leq i \leq min\_sup} \tau_i \qquad (12)$$

Proposition 2 (*Property of $wlbocc$* [8]). $wlbocc$ is a WLB on occ, i.e. $wlbocc(X) \leq occ(S)$ for any strictly frequent itemset X and its frequent proper FE $S = X \oplus Y, Y \neq \emptyset$.

Example 7. For $min\_sup = 2$ and $min\_uo = 0.1$, consider itemset $X = a$, and a proper FE S of X, $X \subset S = X \oplus e = ae$. We have $h(X) = \{T_1, T_4, T_5\}$, $\mathbb{Q}(X,T_1) = 16 + \min\{16\} = 32$, $\mathbb{Q}(X,T_4) = 16 + \min\{10;2;12;52\} = 18$, and $\mathbb{Q}(X,T_5) = 48 + \min\{18;55\} = 66$. Then, $\mathbb{Q}_{rel}(X,T_1) = \mathbb{Q}(X,T_1)/tu(T_1) = \frac{32}{32} = 1$, $\mathbb{Q}_{rel}(X,T_4) = \mathbb{Q}(X,T_4)/tu(T_4) = \frac{18}{92} = 0.196$, $\mathbb{Q}_{rel}(X,T_5) = \mathbb{Q}(X,T_5)/tu(T_5) = \frac{66}{121} = 0.545$ and $\mathcal{F} = \{1; 0.196; 0.545\}$. Thus, $\mathcal{F}' = \{\tau_1, \tau_2, \tau_3\} = \{0.196; 0.545; 1\}$ and $wlbocc(X) = \frac{\tau_1+\tau_2}{min\_sup} = \frac{0.196+0.545}{2} = 0.371 < occ(S) = 0.642$.

Consider an itemset S being evaluated during the mining process, such as $C = bc$, which is formed by joining $X = b$ and $Y = c$. The key challenge is determining whether the $PBr(C)$ branch contains no CFHUOIs and/or GFHUOIs, allowing it to be pruned. This decision relies on elements in the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets, as well as information from previous levels of the prefix tree, including nodes representing X, Y, and their parents. To address this, the paper leverages the following propositions. Notably, by the time $C = bc$ is evaluated, the exploration of $Br(a)$ has already been completed. This prior exploration helps identify and store numerous CFHUOI and GFHUOI candidates in their respective sets, derived from $Br(a)$. Building on theoretical results from [7] and [8], this paper extends the pruning strategies for $NonC\_FHUO$ or $NonG\_FHUO$ branches, as detailed below.

Proposition 3 (*Early pruning of $NonC\_FHUO$ or $NonG\_FHUO$ branches using backwards*). Consider the current itemset $C \in \mathcal{FHUOI}$ and two sets $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ that include the current candidates of CFHUOIs and GFHUOIs, respectively.

(i) (*EPNonCloBr - The halting of the search for CFHUOIs*) The entire branch $Br(C)$ cannot contain any CFHUOIs, i.e. it is a $NonC\_FHUO$ branch, if there exists an itemset $D \in \mathcal{CFHUOI}$ such that $D \supset C, lastItem(C) = lastItem(D)$ and $supp(D) = supp(C)$. Thus, it is unnecessary to find CFHUOIs in this branch.

(ii) (*EPNonGenBr - The halting of the search for GFHUOIs*) The branch $PBr(C)$ cannot include any GFHUOIs, i.e. it is a NonG_FHUO branch, if there exists an itemset $R \in \mathcal{GFHUOI}$ satisfying the following conditions: $R \subset C, supp(R) = supp(C)$ and $wlbocc(R) \geq min\_uo$. Therefore, we can omit the search for larger GFHUOIs generated from C.

The primary limitation of the EPNonCloBr and EPNonGenBr strategies is the need to verify inclusion relationships between itemsets, which can be computationally expensive. To address this, the paper introduces alternative pruning strategies, derived from the theoretical results in [7] and [8], as outlined in the following proposition.

Proposition 4 (*Local pruning of $NonC\_FHUO$ or $NonG\_FHUO$ branches using nodes at two successive levels of the prefix tree*). Consider two nodes $X = P \oplus x$ and $Y = P \oplus y$, which share the same non-empty parent P in the prefix tree, where $y \succ x$, and a forward extension S of X with y, $S = X \oplus y$.

(i) (*LPNonCloBr*) If $supp(S) = supp(Y)$, then $Br(Y)$ is a NonC_FHUO branch and thus we ignore discovering CFHUOIs in this branch.

(ii) (*LPNonGenBr*) If the following conditions are met: ($supp(S) = supp(Y)$ and $wlbocc(Y) \geq \min\_uo$) or ($supp(S) = supp(X)$ and $wlbocc(X) \geq min\_uo$), then $PBr(S)$ is a NonG_FHUO branch, and thus we stop the search for larger GFHUOIs generated from S. Additionally, the non-generator FHUO node corresponding to S is discarded if any of the following conditions hold: $occ(Y) \geq min\_uo, occ(X) \geq min\_uo$, or $S \notin \mathcal{FHUOI}$.

Pruning the $NonC\_FHUO$ $Br(C)$ or $NonG\_FHUO$ $PBr(C)$ branches by Proposition 4 relies solely on the support of itemsets Y (or X) and S without verifying the obvious inclusion $Y \subset S$ (or $X \subset S$, respectively).

One of the main objectives of this paper is to develop an efficient algorithm that simultaneously discovers both $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets. The key difference between the pruning strategies in Propositions 3 and 4 and those in [7] and [8] lies in how the pruning conditions are applied. Specifically, when the conditions in Propositions 3.(i) and 4.(i) (or Propositions 3.(ii) and 4.(ii)) are met, the search halts only for CFHUOIs (or GFHUOIs, respectively). In other words, if the conditions in Propositions 3.(i) and 4.(i) hold, the algorithm continues searching solely for GFHUOIs in $Br(C)$. Similarly, if the conditions in Propositions 3.(ii) and 4.(ii) are satisfied, the algorithm proceeds to mine solely CFHUOIs in $Br(C)$.

3. Novel pruning strategies

Note that the techniques in Propositions 3 and 4 are used to prune only $NonC\_FHUO$ or $NonG\_FHUO$ branches. In the following, the paper introduces novel pruning strategies to prune branches early that do not contain both CFHUOIs and GFHUOIs.

Corollary 1 (*EPNonCGBr - Early pruning of $NonCG\_FHUO$ branches using backwards*). Consider the current itemset $C \in \mathcal{FHUOI}$ and two sets $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ that include the current candidates of CFHUOIs and GFHUOIs, respectively. If there exist itemsets $D \in \mathcal{CFHUOI}$ and $R \in \mathcal{GFHUOI}$ such that the following conditions are met $R \subset C \subset D$, $lastItem(C) = lastItem(D)$, $supp(D) = supp(C) = supp(R)$, and $wlbocc(R) \geq min\_uo$, then the $PBr(C)$ branch cannot include any CFHUOIs and GFHUOIs, i.e. it is a $NonCG\_FHUO$ branch, and thus it can be pruned early.

Proof: Corollary 1 follows directly from Proposition 3. ■

Corollary 2 (*LPNonCGBr - Local pruning of $NonCG\_FHUO$ branches using nodes at three successive levels of the prefix tree*). For any two nodes having the same parent P in the prefix tree, $X = P \oplus x$ and $Y = P \oplus y$, where $y \succ x$, and an itemset $D = prefix(P) \oplus y$, consider a forward extension itemset S of X with the last item y of the node $Y, S = X \oplus y = P \oplus x \oplus y$. Then, if ($supp(S) = supp(P)$ and $wlbocc(P) \geq min\_uo$) or ($supp(S) = supp(D)$ and $wlbocc(D) \geq min\_uo$, then the $NonCG\_FHUO$ $PBr(Y)$ proper branch can be pruned early, since it cannot contain any CFHUOIs and GFHUOIs. In addition, if $occ(P) \geq min\_uo$, $occ(D) \geq min\_uo$ or $Y \notin \mathcal{FHUOI}$, then the Y node is also eliminated safely.

Proof: Since $X = P \oplus x \subset S = P \oplus x \oplus y = P \oplus xy$ and $P \subset Y = P \oplus y \subset S = P \oplus xy$, and $D = parent(P) \oplus y \subset Y = P \oplus y \subset S$, the itemsets S, Y and D share the same last item y. ■

- If $supp(S) = supp(P)$, then by Lemma 1 in [8], $supp(S) = supp(Y) = supp(P)$. By Proposition 4.($i$), $PBr(Y)$ cannot contain any CFHUOIs. In addition, if $wlbocc(P) \geq min\_uo$, then by Proposition 4.(ii), $PBr(Y)$ cannot also contain any GFHUOIs. Therefore, $PBr(Y)$ is a $NonCG\_FHUO$ branch, and thus it can be early pruned.
- Similarly, since $D = parent(P) \oplus y$, we have $D \subset Y \subset S$, the itemsets S and D share the same last item y. If $supp(S) = supp(D)$, then by Lemma 1 in [8], we have $supp(S) = supp(Y) = supp(D)$. If $wlbocc(D) \geq min\_uo$, the whole $NonCG\_FHUO$ $PBr(Y)$ proper branch is also pruned safely by Proposition 4.

The remaining assertions are proven similarly.

Example 8. Figure 1 illustrates applying the pruning techniques outlined in Proposition 4 and Corollary 2 to prune $NonC\_FHUO$ and/or $NonG\_FHUO$ branches in the prefix tree. The figure presents a portion of the tree generated using the FE operator with parameters $min\_uo = 0.18$ and $min\_sup = 1$. To improve clarity and readability, each node is labeled with an itemset rather than just its last item. Additionally, four values are assigned to each node, representing support ($supp$) and utility occupancy (occ) as superscripts, and the weak upper bound ($wubocc$) and weak lower bound ($wlbocc$) as subscripts. This notation provides a clearer and more intuitive representation of the example. For instance, the itemset $bc^{2;0.11}_{1;0.26}$ signifies that $supp(bc) = 2$, $occ(bc) = 0.11$, $wubocc(bc) = 1$ and $wlbocc(bc) = 0.26$. Consider the itemsets $X = bc^{2;0.11}_{1;0.26}$ and $Y = bd^{2;0.25}_{0.92;0.54}$, which share the same prefix $P = b^{2;0.06}_{1;0.09}$, along with the itemset $D = d^{2;0.19}_{0.9;0.53}$. When X is extended by adding d, forming a new itemset $S = X \oplus d = bcd^{2;0.3}_{1;0.63}$, it is observed that $supp(S) = supp(Y) = supp(D) = 2$ and $wlbocc(D) = 0.53 \geq min\_uo$. As a result, applying the $LPNonCGBr$ strategy from Corollary 2 enables early pruning of the $NonCG\_FHUO$ branch $PBr(Y)$ (highlighted with a red border) since it is guaranteed to contain no any CFHUOIs and GFHUOIs. Furthermore, since $occ(D) = 0.19 > min\_uo$, and despite Y being an FHUOI, the Y node, which is neither a CFHUOI nor a GFHUOI, is also pruned, along with the $NonCG\_FHUO$ branch $Br(Y)$. Similarly, when X is extended with e, forming the itemset $S_1 = X \oplus e = bce^{2;0.54}_{0.76;0;76}$, it is observed that $supp\,(S_1) = supp(be) = 2$ and $wlbocc(be) = 0.67 > min\_uo$. Consequently, according to Proposition 4, this enables early pruning of both the $NonC\_FHUO$ $Br(be)$ and $NonG\_FHUO$ $PBr\,(S_1)$ (highlighted with a blue border). Additionally, although S_1 belongs to the $\mathcal{FHUOI}$ set, since $occ(be) = 0.49 > min\_uo$, the $NonG\_FHUO$ S_1 node is also eliminated.

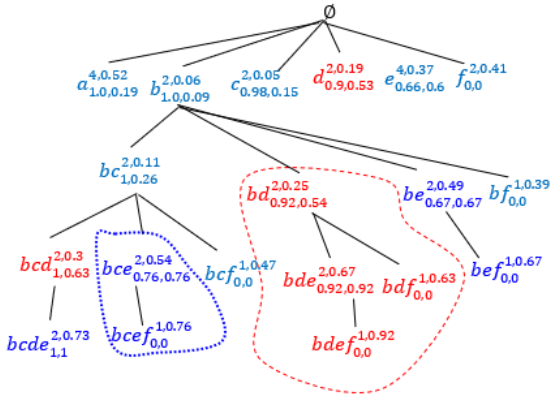


Figure 1. Illustrating strategies in Proposition 4 and Corollary 2.

4. Forming the result sets

a) The construction of the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets

After eliminating all unpromising patterns, those that are infrequent, have low utility occupancy, or are non-closed and non-generator FHUOIs, using Propositions 1-2 and Corollaries $1-2$, the remaining itemsets become strong candidates for CFHUOIs and GFHUOIs.

Let the search space for mining the $\mathcal{CFHUOI}$ set be represented as a prefix tree, where a depth-first search is performed following the predefined item order $\prec$. Initially, $\mathcal{CFHUOI}$ is first initialized as an empty set. For a CFHUOI candidate C being under consideration, it will be added to $\mathcal{CFHUOI}$, if the following condition (1) is satisfied:

$$[C \in \mathcal{FHUOI}, C \text{ has no } 1\text{- forward } and(\nexists D \in \mathcal{CFHUOI} : D \supset C \text{ and } supp(D) = supp(C))] \quad (13)$$

Note that if D is a $1-$ forward or $1-$ backward extension of C, it will either be processed after C or has already been processed before C, respectively, during prefix tree construction. For example, given $C = ce$, its $1-$forward extension is cef, while cde and bce are its $1-$backward extensions. This follows from their equal support values: $supp(cef) = supp(cde) = supp(bce) = supp(C)$. Based on (1), the following proposition [7] is used to construct the $\mathcal{CFHUOI}$ set efficiently.

Proposition 5 (*The construction of the $\mathcal{CFHUOI}$ set* [7])

$$\mathcal{CFHUOI} = \{C \in \mathcal{A} \mid C \text{ is a } CFHUOI\}$$

This proposition states that by adding itemsets C to $\mathcal{CFHUOI}$ based on the conditions in (1), the $\mathcal{CFHUOI}$ set is constructed correctly and completely. Moreover, the aforementioned process for the form of $\mathcal{CFHUOI}$ ensures that no non-closed FHUOIs are generated during the process.

For the construction of the $\mathcal{CFHUOI}$ set, which is defined as $\mathcal{GFHUOI} \stackrel{\text{def}}{=} \{A \in \mathcal{FHUOI} \mid \nexists B \in \mathcal{FHUOI} : B \subset A \wedge supp(B) = supp(A)\}$, it requires checking two conditions: (1) A belongs to the $\mathcal{FHUOI}$ set, and (2) no proper subset B of A in $\mathcal{FHUOI}$ shares the same support as A. During the mining process, each remaining itemset S is evaluated. If S is an $FHUOI$ and no itemset in $\mathcal{CFHUOI}$ is its proper subset with the same support, S is added to proper. Then, for each candidate R in $\mathcal{GFHUOI}$, if $S \subset R$ and $supp(R) = supp(S), R$ is removed since it cannot be a GFHUOI. This iterative process ensures that the $\mathcal{GFHUOI}$ set is continuously updated throughout mining. Upon completion, the remaining itemsets in $\mathcal{GFHUOI}$ constitute the final and complete set of GFHUOIs.

b) The construction of the $\mathcal{MFHUOI}$ set

As mentioned in the Introduction section, the paper aims to develop two effective algorithms to simultaneously discover both CFHUOIs and GFHUOIs at the same process, and to exploit MFHUOIs separately due to their benefits. For CFHUOIs and GFHUOIs, we use the theoretical results in Sections 4.2-4.3 to eliminate unpromising candidates, and the method in Section 4.4.a to construct the result sets $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$. For MFHUOIs, new methods are required to discover them effectively. Note that we have the following assertion $\mathcal{MFHUOI} \subseteq \mathcal{CFHUOI} \subseteq \mathcal{FHUOI}$, and thus in [7] the authors introduced the method for determining the $\mathcal{MFHUOI}$ set from $\mathcal{CFHUOI}$ as follows: $\mathcal{MFHUOI} = \{A \in \mathcal{CFHUOI} \mid \nexists B \in \mathcal{CFHUOI} : B \supset A\}$. However, since the cardinality of the $\mathcal{CFHUOI}$ set is often much larger than that of $\mathcal{MFHUOI}$, checking the inclusive relationship between a larger number of itemsets may take much more time. In addition, as the current study needs to develop a new algorithm that discovers only $\mathcal{MFHUOI}$ without $\mathcal{CFHUOI}$, constructing the $\mathcal{CFHUOI}$ set is unnecessary. Based on Proposition 3.(i), Proposition 4.(i) and the relationship $\mathcal{MFHUOI} \subseteq \mathcal{CFHUOI}$, the following corollary is derived.

Corollary 3 (*NonMaxBr Pruning strategies*).

(i) (*EPNonMaxBr - Early pruning of NonM_FHUO branches by checking backwards*) Consider the current itemset $C \in \mathcal{FHUOI}$ and the $\mathcal{MFHUOI}$ set of MFHUOI candidates. If there exists an itemset $D \in \mathcal{MFHUOI}$ such that $D \supset C$, $lastItem(C) = lastItem(D)$ and $supp(D) = supp(C)$, the entire branch $Br(C)$

cannot contain any MFHUOIs, and thus it can be pruned early.

(ii) (*LPNonMaxBr - Local pruning of NonM_FHUO branches*) Consider two nodes $X = P \oplus x$ and $Y = P \oplus y$, which share the same non-empty parent P in the prefix tree, where $y \succ x$, and a forward extension S of X with y, $S = X \oplus y$. If $supp(S) = supp(Y)$, then the $NonM\_FHUO$ $Br(Y)$ branch can be eliminated early.

Based on the definition of the $\mathcal{MFHUOI}$ set, it can be constructed by verifying two conditions: $X \in \mathcal{FHUOI}$ and $(\nexists Y \in \mathcal{FHUOI} : Y \supset X)$. For each remaining FHUOI S that is not pruned by Corollary 3 in the mining process, if no itemset in $\mathcal{MFHUOI}$ is its superset, it will be added to the $\mathcal{MFHUOI}$ set. Next, for each candidate itemset R in $\mathcal{MFHUOI}$, if the condition $R \subset S$ holds, R is removed from $\mathcal{MFHUOI}$. Hence, the $\mathcal{MFHUOI}$ is continuously updated during the mining process. When the algorithm finishes, the complete and correct $\mathcal{MFHUOI}$ set of all MFHUOIs is constructed.

Example 9. Consider two itemsets $X = bcd^{2;0.3}_{1;0.63}$ and $Y = bce^{2;0.54}_{0.76;0.76}$ in Figure 1. When X is extended with the last item e of Y to create a new itemset $S = X \oplus y = bcde^{2;0.73}_{1;1}$, it is observed that the pruning condition $supp(S) = supp(Y) = 2$ is satisfied. Thus, the whole $Br(Y)$ branch cannot contain any MFHUOIs, i.e. it is a NonM_FHUO branch, and it can be eliminated early to reduce the search space.

5. Proposed algorithms

Based on the proposed theoretical results, this section develops two novel algorithms, which are CGFHUOI-Miner to simultaneously discover both CFHUOIs and GFHUOIs in the same process, and MaxFHUOI-Miner to exploit MFHUOIs. The proposed algorithms are implemented using the MUO-list data structure, which is a modified version of UO-list in [19] to additionally store necessary information needed for discovering concise representations of FHUOIs. **Figure 2** illustrates the Pre-processing procedure, which operates on a QTDB $\mathcal{D}'$ as input and requires two threshold parameters min_uo and min_sup. Initially, the procedure scans the $\mathcal{D}'$ to compute the support of each item and the total utility (tu) value of each transaction. Based on this computation, it identifies the set E of frequent items, where each item's support is greater than or equal to min_sup (line 1). Infrequent items are then removed from $\mathcal{D}'$ to reduce QTDB, thus facilitating the subsequent item-extension process (line 2). Following this, the procedure sorts the remaining items in ascending order based on their supports to enhance the efficiency of further operations.

Procedure 1: Pre-processing.
Input: QTDB $\mathcal{D}'$, min_uo and min_sup.
Output: + Two sets $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ of all CFHUOIs and GFHUOIs if using the CGFHUOI-Miner algorithm, or
+ The $\mathcal{MFHUOI}$ set of all MFHUOIs if using the MaxFHUOI-Miner algorithm.

1. **Scan** $\mathcal{D}'$ to calculate the tu value of each transaction, and identify E, the list of frequent 1-itemsets
2. **Remove** all *infrequent* items from $\mathcal{D}'$;
3. **Scan** $\mathcal{D}'$ again to construct the structure MUO-list for all *frequent* items in E;
4. $\mathcal{CFHUOI} := \emptyset$; $\mathcal{GFHUOI} := \emptyset$; $\mathcal{MFHUOI} := \emptyset$;
5. **for** each $x \in E$ **do**
6. a. **CGFHUOI-Miner** (x, E, min_uo, min_sup, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$);
6. b. **MaxFHUOI-Miner** (x, E, min_uo, min_sup, $\mathcal{MFHUOI}$);
7. **return** ($\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$), or $\mathcal{MFHUOI}$;

Figure 2. The Pre-processing procedure.

The algorithm scans $\mathcal{D}'$ once to construct the MUO-list structure for all items in E (line 3). For each frequent item x in E, it calls the CGFHUOI-Miner algorithm (line 6.a) to simultaneously discover both CFHUOIs and GFHUOIs at the same process, or invokes the MaxFHUOI-Miner algorithm (line 6.b) to exploit only MFHUOIs. Upon completion, the procedure outputs the discovered sets, $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$, or $\mathcal{MFHUOI}$ (line 7).

The CGFHUOI-Miner algorithm. Figure 3 shows the CGFHUOI-Miner algorithm that begins by evaluating whether any pruning conditions (lines 1 and 2) are satisfied for the itemset X. If the condition in line 1 is met, the corresponding branch $Br(X)$ is pruned, and the algorithm terminates for X. If the condition in line 2 holds, the algorithm also stops after completing the procedures Update-GenFHUOI and Update-CloFHUOI if X is identified as an FHUOI (line 3). Next, strategies in Corollaries 1-2 are used to early prune $NonCG\_FHUO$ branches that cannot includes both CFHUOIs and GFHUOIs (lines 9-13). If these branches are not eliminated, the algorithm will use strategies in Propositions 3 and 4. At lines 14-19, Propositions 3.(ii) and 4.(ii) are applied to early eliminate $NonG\_FHUO$ proper branch $PBr(X)$, while Propositions 3.(i) and 4.(i) are adopted in lines 20-23 to prune $NonC\_FHUO$ branch $Br(X)$. Note that if $X.Prune\_NonG\_PBr$ is true (line 14), the algorithm only ignores discovering GFHUOIs in $PBr(X)$, but mining

Algorithm 1: CGFHUOI-Miner (X, E, min_uo, min_sup, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$)

Input: The itemset X, set E, min_uo, min_sup, and two sets $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$.

Output: $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ have been updated.

1. **if** ($supp(X) < min\_sup$) **then return;** // *Pruning //strategy based on support*
2. **if** ($wubocc(X) < min\_uo$) **then {** // *Strategy EPLUOI*
3. **if** ($occ(X) \geq min\_uo$) **then {**
4. Update-GenFHUOI (X, min_uo, $\mathcal{GFHUOI}$);
5. Update-CloFHUOI (X, min_uo, $\mathcal{CFHUOI}$);
6. **}**
7. **return**;
8. **}**
9. **if** ($X.Prune\_NonCG\_Br$ **or** ($X.Prune\_NonC\_Br$ **and** $X.Prune\_NonG\_Br$) **or** ***hasBackwardExt_CG***(X, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$)) **then** **{** // *Corollary 2 & Corollary 1*
10. **if** ($X.Prune\_NonG\_Node = false$) **then**
11. Update-GenFHUOI (X, min_uo, $\mathcal{GFHUOI}$);
12. **return;**
13. **}**
14. **if** ($X.Prune\_NonG\_PBr$) **then** { *//Propositions 3 and 4.(ii)*
15. **If** ($X.Prune\_NonG\_Node = false$) **then**
16. Update-GenFHUOI (X, min_uo, $\mathcal{GFHUOI}$);
17. **Search-CloFHUOI** (X, E, min_uo, min_sup, $\mathcal{CFHUOI}$);
18. **return**;
19. **}**
20. **If** ($X.Prune\_NonC\_Br$) **then** { *//Propositions 3 and 4.(i)*
21. **Search-GenFHUOI** (X, E, min_uo, min_sup, $\mathcal{GFHUOI}$);
22. **return**;
23. **}**
24. $P = prefix(X)$; $ListEx = \emptyset$;
25. **for each** $y \in E$ such that $y \succ lastItem(X)$ **do** {
26. **if** ($supp(Xy) \geq min\_sup$) **then** {
27. **Add** $S = X \oplus y$ to $ListEx$ and build S'MUO-list;
28. **EarlyPruning-NonCGI** (X, Py, S); // *Proposition 4 and Corollary 2*
29. **}**
30. **}**
31. **if** ($occ(X) \geq min\_uo$) **then** {
32. Update-GenFHUOI (X, min_uo, $\mathcal{GFHUOI}$);
33. Update-CloFHUOI (X, min_uo, $\mathcal{CFHUOI}$);
34. **}**
35. **for each** $S \in ListEx$ **do**
36. **CGFHUOI-Miner**(S, E, min_uo, min_sup, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$);
37. **return** $\mathcal{CFHUOI}$ **and** $\mathcal{GFHUOI}$;

Figure 3. The CGFHUOI-Miner algorithm.

CFHUOIs in this branch is still required by invoking the Search-CloFHUOI proceduce, which is similar to the FindMaxCloFHUOI procedure presented in [7]. Similarly, if $X.Prune\_NonC\_Br = true$ (line 20), mining CFHUOIs in the $Br(X)$ branch is omitted, but the Search-GenFHUOI procedure [8] is called to exploit GFHUOIs in this branch. Next, for each item $y \in E$ such that $y \succ lastItem(X)$, CGFHUOI-Miner constructs the MUO-list structure for the extension $S = Xy$ and calls the EarlyPruning-NonCGI procedure to mark for early pruning branches based on conditions shown in Proposition 4 and Corollary 2 (lines 25-30). After that, if the condition $occ(X) \geq min\_uo$ in line 31 holds, the procedures Update-GenFHUOI and Update-CloFHUOI are called to update the sets $\mathcal{GFHUOI}$ and $\mathcal{CFHUOI}$, respectively (lines 32 and 33). Finally, for each itemset S in ListEx, the CGFHUOI-Miner algorithm recursively calls itself (lines 35-36) to explore and identify larger CFHUOIs and GFHUOIs.

The MaxFHUOI-Miner algorithm. The pseudo-code of the MaxFHUOI-Miner algorithm is shown in Figure 4. Like the CGFHUOI-Miner algorithm, MaxFHUOI-Miner takes as input an itemset X, the E set of candidate items for extending X, two minimum thresholds min_uo and min_sup, and the $\mathcal{MFHUOI}$ set of promising patterns for MFHUOIs. Its output is the complete and correct $\mathcal{MFHUOI}$ set of all MFHUOIs. First, the algorithm begins with checking the pruning conditions at line 1 , and it will stop the mining process on the subtree rooted at the X node if one of the conditions is satisfied according to Corollary 3. (i). Then, if the $wubocc$ value of X is less than min_uo, the Update-MaxFHUOI procedure is called (if the condition $occ(X) \geq min\_uo$ holds) before discovering MFHUOIs in the $Br(X)$ branch is terminated.

Next, for each item y belonging to the E set such that $y \succ lastItem(X)$, if itemset $S = Xy$ is frequent, the algorithm constructs the data structure MUO-list of S. Then, it marks the $Br(Py)$ branch for early pruning if the $supp(S) = supp(Py)$ condition holds. Finally, for each itemset S belonging to the ListEx set, the MaxFHUOI-Miner algorithm recursively calls itself to discover larger MFHUOIs in the $Br(X)$ branch.

Remark (*Optimization strategy: Reducing subset checks and repeated candidate verifications in the sets* $\mathcal{CFHUOI}$ *and* $\mathcal{GFHUOI}$).

To enhance the efficiency of updating the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets, we propose an optimization strategy that minimizes redundant subset checks and repeated candidate verifications. Instead of employing support values as hash keys, an approach commonly adopted in prior studies, we utilize the TWU values of itemsets to index $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ via hash tables. Here,

Algorithm 2: MaxFHUOI-Miner (X, E, min_uo, min_sup, $\mathcal{MFHUOI}$)
Input: The itemset X, the set E, thresholds min_uo and min_sup, and $\mathcal{MFHUOI}$.
Output: The set $\mathcal{MFHUOI}$ has been updated.

1. **if** ($supp(X) < ms$ **or** $X.isPruned$ **or** *// Corollary 3. (i)*
   hasBackwardExt_Max (X, $\mathcal{MFHUOI}$)) **then**
2. **return**;
3. **if** ($wubocc(X) < min\_uo$) **then** {*// Strategy EPLUOI*
4. **if** ($occ(X) \geq min\_uo$) **then**
   Update-MaxFHUOI(X, $\mathcal{MFHUOI}$);
5. **return**;
6. }
7. $P = prefix(X)$; $ListEx = \emptyset$;
8. **for each** $y \in E$ such that $y \succ lastItem(X)$ **do** {
9. **if** ($supp(Xy) \geq min\_sup$) **then** {
10. **Add** $S = Xy$ to $ListEx$ and construct the structure MUO-list of S;
11. **if** ($supp(S) = supp(Py)$) **then**
12. $Py.isPruned = true$; *//Corollary 3. (ii)*
13. }
14. }
15. **if** ($occ(X) \geq min\_uo$) **then**
16. **Update-MaxFHUOI**(X, $\mathcal{MFHUOI}$);
17. **for each** $S \in ListEx$ **do**
18. **MaxFHUOI-Miner** (S, $newE$, min_uo, ms, $\mathcal{MFHUOI}$);
19. **return** $\mathcal{MFHUOI}$;

Figure 4. The MaxFHUOI-Miner algorithm.

Procedure 2: EarlyPruning-NonCGI (X, Y, S)
Input: Three itemsets X, Y, and S.
Output: Two itemsets Y and S have been updated.

1. $P = prefix(X)$; $D = prefix(P) \oplus lastItem(Y)$;
2. **if** (($supp(S) = supp(P)$ **and** $wlbocc(P) \geq min\_uo$) **or** ($supp(S) = supp(D)$ **and** $wlbocc(D) \geq min\_uo$)) **then** {
3. $Y.Prune\_NonCG\_Br = true$; *//Corollary 2*
4. **if** ($occ(P) \geq min\_uo$ **or** $occ(D) \geq min\_uo$ **or** $occ(Y) < min\_uo$) **then**
5. $Y.Prune\_NonG\_Node = true$; *//Corollary 2*
6. }
7. **else** {
8. **if** ($supp(S) = supp(Y)$)
9. $Y.Prune\_NonC\_Br = true$; *//Proposition 4. (i)*
10. }
11. **if** (($supp(S) = supp(Y)$ **and** $wlbocc(Y) \geq min\_uo$) **or** ($supp(S) = supp(X)$ **and** $wlbocc(X) \geq min\_uo$)) **then {**
12. $S.Prune\_NonG\_PBr = true$; *//Proposition 4. (ii)*
13. **if** ($occ(Y) \geq min\_uo$ **or** $occ(X) \geq min\_uo$ **or** $occ(S) < min\_uo$) **then**
14. $S.Prune\_NonG\_Node = true$; *//Proposition 4. (ii)*
15. }

Figure 5. The EarlyPruning-NonCGI procedure.

the TWU of an itemset X is defined as $TWU(X) = \sum_{T_i \in \rho(X)} tu(T_i)$, where $\rho(X)$ denotes the set of transactions containing X, and $tu(T_i)$ is the total utility of transaction T_i. This choice is grounded in the following insight: if an itemset X fails to satisfy the conditions for inclusion in $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$, namely, if there exists an itemset $\exists Y \in \mathcal{FHUOI}$ such that $Y \subset X$ (or $Y \supset X$) and $supp(Y) = supp(X)$, then it must hold that $\rho(Y) = \rho(X)$ and consequently $TWU(Y) = TWU(X)$. As a result, X and Y will reside in the same bucket of the TWU-indexed hash table.

Compared to support-based indexing, TWU-based hashing significantly reduces hash collisions. For example, in a scenario where $min\_uo = 0.18$ and $min\_sup = 1$, the number of discovered FHUOIs is 46. Consider itemset $X = af$, which appears in only one transaction T_5, resulting in $supp(X) = 1$ and $TWU(X) = 121$. Then, only two FHUOIs, af and aef, share the same TWU value, whereas as many as 29 FHUOIs have a support value of 1. This illustrates that TWU provides a finer granularity for distinguishing itemsets, thus reducing unnecessary comparisons.

Example 10 (*An illustration of the main steps of the CGFHUOI-Miner algorithm*).

An illustrative example demonstrating key steps of the CGFHUOI-Miner algorithm is presented in the following figures, with parameters set to $min\_uo = 0.18$ and $min\_sup = 1$, where items are arranged in lexicographic order. Initially, Procedure 1 is executed to preprocess the database $\mathcal{D}'$ by scanning each transaction to compute its transaction utility (tu) and identify the set E of valid items. As all items satisfy the minimum support threshold, no items are removed from $\mathcal{D}'$ (line 2 of Procedure 1). Subsequently, for each item $x \in E = \{a, b, c, d, e, f\}$, the CGFHUOI-Miner algorithm is invoked to discover CFHUOIs and GFHUOIs from the corresponding branch $Br(x)$, which includes x and its forward extensions. In the execution trace for branch $Br(A)$ illustrated in Figure 8, none of the pruning conditions at lines $1, 2, 9, 14$, and 20 of Algorithm 1 are satisfied. Consequently, item x is extended with items b, c, d, e, f to generate the $2-$itemsets ab, ac, ad, ae, af (lines 25-29), which are not pruned by the condition at line 26. Furthermore, when the EarlyPruningNonCGI procedure is invoked at line 28

Procedure 3: Update-CloFHUOI (C, $\mathcal{CFHUOI}$)
Input: The FHUOI C and the set $\mathcal{CFHUOI}$.
Output: $\mathcal{CFHUOI}$ has been updated.
1. **for** each $D \in \mathcal{CFHUOI}$ **do**
2. **if** ($C \subset D$ and $supp(C) = supp(D)$) **then**
3. **return**;
4. **for** each $R \in \mathcal{CFHUOI}$ **do**
5. **if** ($R \subset C$ and $supp(R) = supp(C)$) **then**
6. *Remove* R from $\mathcal{CFHUOI}$;
7. $\mathcal{CFHUOI} \coloneqq \mathcal{CFHUOI} \cup \{C\}$;

Procedure 4: Update-GenFHUOI (C, $\mathcal{GFHUOI}$)
Input: The FHUOI C and the set $\mathcal{GFHUOI}$.
Output: $\mathcal{GFHUOI}$ has been updated.
1. **for** each $D \in \mathcal{GFHUOI}$ **do**
2. **if** ($D \subset C$ and $supp(C) = supp(D)$) **then**
3. **return;**
4. **for** each $R \in \mathcal{GFHUOI}$ **do**
5. **if** ($C \subset R$ and $supp(R) = supp(C)$) **then**
6. Remove R from $\mathcal{GFHUOI}$;
7. $\mathcal{GFHUOI} \coloneqq \mathcal{GFHUOI} \cup \{C\}$;

Figure 6. The Update-CloFHUOI and Update-GenFHUOI procedures.

Procedure 5: *hasBackwardExt_CG*(C, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$)
Input: The itemset C and two sets $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$.
Output: the *True* value is returned, if C has a superset in $\mathcal{CFHUOI}$ and a subset in $\mathcal{GFHUOI}$ with the same support; otherwise, it returns *False*.
1. **for** each $D \in \mathcal{CFHUOI}$ **and** $R \in \mathcal{GFHUOI}$ **do** *//Corollary 1*
2. **if** ($R \subset C \subset D$ **and** $lastItem(C) = lastItem(D)$ **and** $supp(C) = supp(D) = supp(R)$ **and** $wlbocc(R) \geq min\_uo$) **then**
3. **return** true; *//Corollary 1*
4. **return** false;

Figure 7. The hasBackwardExt_CG procedure.

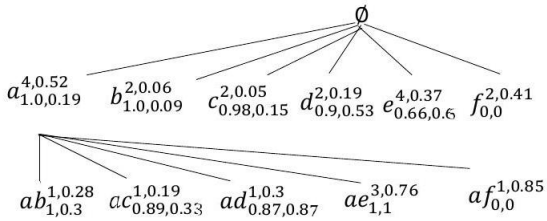


Figure 8. Execution of CGFHUOI-Miner on the $Br(a)$ branch.

for each generated 2−itemset, no branches are pruned in accordance with Proposition 4 and Corollary 2. Since the condition $occ(X) \geq min\_uo$ is satisfied (line 31), item X is added to both the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets through the Update-CloFHUOI and Update-GenFHUOI procedures (lines 32-33), resulting in $\mathcal{CFHUOI} = \{a\}$ and $\mathcal{GFHUOI} = \{a\}$.

Next, the execution of CGFHUOI-Miner for the branch $Br(X)$, $X = ab$, is examined in Figure 9. As the pruning conditions at lines 1, 2, 9, 14, and 20 in Algorithm 1 are not satisfied, the $Br(X)$ branch cannot be pruned and is retained for further exploration. When X is extended with the last item $y = c$ from itemset $Y = ac$, resulting in the new itemset $S = X \oplus y = abc$ (line 27), Procedure 2 (EarlyPruning-NonCGI) in Figure 5 is invoked to determine whether any branches can be eliminated early based on the proposed pruning strategies. Within Procedure 2, it is found that $supp(S) = supp(Y) = 1$ (line 8) and $wlbocc(Y) = 0.33 \geq min\_uo$ (line 11). Hence, the discovery of CFHUOIs in the $Br(Y)$ branch is skipped (line 9) according to Proposition 4. (i), and the branch is marked for early pruning in subsequent calls to CGFHUOI-Miner. Furthermore, since $occ(Y) = 0.19 > min\_uo$ (line 13), the branch $Br(S)$ is also marked for pruning as a $NonG\_FHUO$ branch, following Proposition 4. (ii) (lines 12 and 14). Similarly, when X is extended with items d and e, producing the itemsets abd and abe, the corresponding branches $Br(abd)$ and $Br(ad)$ are marked as $NonG\_FHUO$ and $NonC\_FHUO$, respectively, and are pruned in accordance with Propositions 4. (i) and 4. (ii). It is also noted that the itemset abf is not generated, as it does not occur in any transaction (i.e., $supp(abf) = 0$). Finally, since ab satisfies the required conditions, it is added to both $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ by lines 32−33 of Algorithm 1, resulting in the updated sets $\mathcal{CFHUOI} = \{a, ab\}$ and $\mathcal{GFHOUI} = \{a, ab\}$.

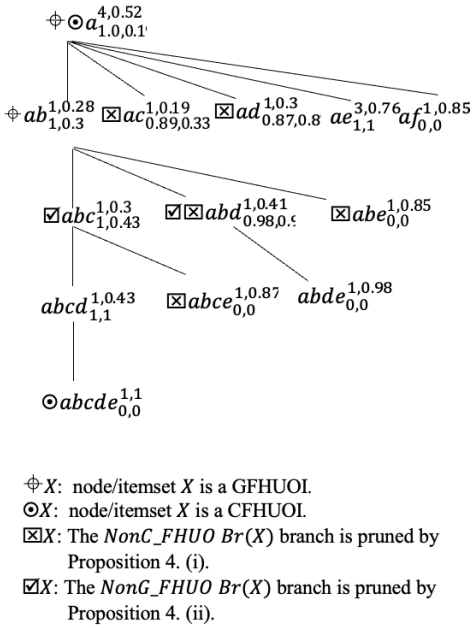


Figure 9. Execution of CGFHUOI-Miner on the $Br(a)$ branch.

We now examine the execution of CGFHUOI-Miner for the itemset $X = abc$, as illustrated in Figure 9. Since the branch $Br(X)$ has been previously marked as containing no GFHUOI, only the Search-CloFHUOI procedure is invoked to identify CFHUOIs within this branch (line 17). When X is extended with items d and e, forming the itemsets $abcd$ and $abce$, Procedure 2 (Figure 5) is called. As a result, the branches $Br(abcd)$ and $Br(abce)$ are marked as NonC_FHUO and pruned early, as they cannot contain any CFHUOIs. Subsequently, since $occ(X) \geq min\_uo$ (line 31), the procedures Update-CloFHUOI and Update-GenFHUOI are executed to update the $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$ sets, respectively (lines 32-33). During the execution of Update-CloFHUOI, the itemset ab is removed from $\mathcal{CFHUOI}$ and abc is added, because $R = ab \subset C = X = abc$ and $supp(R) = supp(C) = 1$, where $R \in \mathcal{CFHUOI}$ (lines 4-7 in Procedure 3). Meanwhile, the $\mathcal{GFHUOI}$ set remains unchanged. The updated sets are thus $\mathcal{CFHUOI} = \{a, abc\}$ and $\mathcal{GFHUOI} = \{a, ab\}$. Similarly, when CGFHUOI-Miner is recursively invoked for $X = abcd$ (see Figure 9), the itemset $abcde$ is generated, and the branch $Br(abce)$ is marked as $NonC\_FHUO$ for early pruning. Then, abc is removed from $\mathcal{CFHUOI}$ and $abcd$ is added via the UpdateCloFHUOI procedure. Following the same procedure, during the execution of CGFHUOI-Miner with $X = abcde$, the itemset $abcd$ is removed from $\mathcal{CFHUOI}$, and $abcde$ is added. At this point, the final sets become $\mathcal{CFHUOI} = \{a, abcde\}$ and

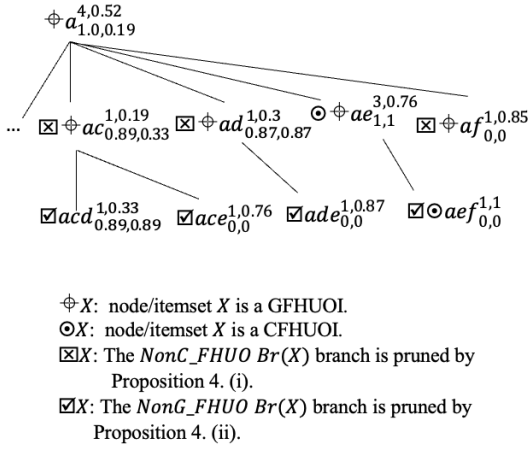


Figure 10. Execution of CGFHUOI-Miner on the $Br(ac), Br(ad), Br(ae)$ and $Br(af)$ branches.

$\mathcal{GFHUOI} = \{a, ab\}$. It is also important to note that the recursive calls to CGFHUOI-Miner for $X = abce$ and $X = abd$ result in no further exploration, as their respective branches have been previously marked as $NonCG\_FHUO$. According to Corollaries 1 and 2, these branches satisfy the condition in line 9 of Algorithm 1 and are therefore pruned early.

Subsequently, during the execution of CGFHUOI-Miner for $X = abe$, followed by $X = ac$ and $X = ad$ (see Figure 10), the branches $Br(X)$ are early pruned (line 20 of Algorithm 1), since they were previously marked as $NonC\_FHUO$ for elimination. As a result, only GFHUOIs need to be discovered in these branches through the invocation of the SearchGenFHUOI procedure (line 21 of Algorithm 1). Given that $supp(abe) = supp(ab) = 1$, the itemset abe is not added to the $\mathcal{GFHUOI}$ set. In contrast, the itemsets ac and ad satisfy the required conditions and are thus added to $\mathcal{GFHUOI}$, resulting in $\mathcal{GFHUOI} = \{a, ab, ac, ad\}$. Furthermore, the branches $Br(acd), Br(ace)$ and $Br(ade)$ are marked for early pruning as $NonG\_FHUO$, according to the LPNonGenBr pruning strategy defined in Proposition 4. (ii). In the subsequent execution of CGFHUOI-Miner for $X = ae$, when X is extended with item f to generate the itemset aef, $Br(aef)$ and $Br(af)$ are marked as $NonG\_FHUO$ and $NonC\_FHUO$, respectively, for early elimination. Consequently, the itemsets ae and aef are added to the $\mathcal{CFHUOI}$ set, while ae and af are included in the $\mathcal{GFHUOI}$ set.

Upon completing the execution of CGFHUOI-Miner for the branch $Br(A)$, the resulting sets are $\mathcal{GFHUOI} = \{a, ab, ac, ad, ae, af\}$ and

$\mathcal{CFHUOI} = \{a, abcde, ae, aef\}$. Similar results are obtained from the execution of CGFHUOI-Miner on the remaining branches $Br(b), Br(C), Br(d), Br(e)$, and $Br(f)$.

Finally, the complete result sets are: $\mathcal{GFHUOI} = \{a, ab, ac, ad, ae, af, be, bf, ce, cf, d, df, e, f\}$ and $\mathcal{CFHUOI} = \{a, abcde, ae, aef, bcde, bcdef, e, ef\}$.

Time complexity of the proposed algorithms. This paragraph begins by analyzing the worst-case time complexity of the MaxFHUOI-Miner algorithm, which is primarily influenced by the number of forward extensions. Let N denote the total number of items in the set $\mathcal{A}$, and let $n = \max\{|T| : T \in \mathcal{D}\}$ be the maximum transaction length. For any itemset X with length $L = |X|$, and where $a_i = lastItem(X)$ and $L \leq i \leq N$, the recursive algorithm MaxFHUOI-Miner$(X, E, min_uo, min_sup, \mathcal{MFHUOI})$, or simply MaxFHUOI-Miner (X), can generate up to $FE(i) \stackrel{\text{def}}{=} N - i$ forward extensions, as indicated in Line 15 of Algorithm 2 in Figure 4. At step L, where $1 \leq L \leq n$, let $T(L, i)$ represent the computational cost of executing MaxFHUOI-Miner (X) for a specific itemset X, and let $\mathcal{T}(L)$ denote the total cost for all itemsets X with length $|X| = L$. Let us define $FHUOI(L)$ as the set of all frequent high utility occupancy itemsets of length L, with the convention that $|FHUOI(0)| = 1$. Additionally, the maximum number of forward extensions possible for all itemsets of length L is given by $ME(L) \stackrel{\text{def}}{=} \sum_{i=L..N} FE(i) = (N - L)(N - L + 1)/2 = O\left(N^2\right)$. Then, $T(L, i) = FE(i) + \sum_{k=i+1..N} T(L + 1, k)$ for $L = 1, \ldots, n-1, \mathcal{T}(L) = |FHUOI(L-1)| \times \sum_{i=L..N} T(L, i)$, and $T(n, i) \stackrel{\text{def}}{=} 0, \mathcal{T}(n) \stackrel{\text{def}}{=} 0$. Moreover, $T(n-1, i) = \text{FE}\ (i) = N - i = O(N - i), \mathcal{T}(n-1) = |FHUOI(n-2)| \times ME(n-1) = |\ FHUOI(n-2)| \times (N-n+1)(N-n+2)/2 = |FHUOI(n-2)\ | \times O\left((N - n + 2)^2\right)$. Similarly, we can derive that $T(n-2, i) = (N-i)(N-i+1)/2 = O\left((N-i)^2\right)$, and the total complexity at level $n-2$ is given by $\mathcal{T}(n-2) = |FHUOI(n-3)| \times [(N-n+2)(N-n+3)(N-n+4)/6] = |FHUOI(n-3)| \times O\left((N - n + 2)^3\right)$. In general, using induction, we obtain $T(L, i) = O\left((N-i)^{n-L}\right)$ and $\mathcal{T}(L) = |FHUOI(L-1)| \times O((N-n+2)^{n-L+1})$. Specifically, the worst-case time complexity of MaxFHUOI-Miner is $\mathcal{T}(1) = O\left((N - n + 2)^n\right)$. By the same reasoning, the worst-case complexity of CGFHUOI-Miner is also $O\left((N - n + 2)^n\right)$. Notably, when $n = N$, we have $\mathcal{T}(1) = 2^N - 1 = O\left(2^N\right)$, which corresponds to the total number of non-empty subsets of the set $\mathcal{A}$ containing N items. However, when early pruning conditions are satisfied (e.g., as in Lines 1, 3, and 11 of Algorithm 2), such as when $wubocc(X) < min\_uo$ in Line 3, all forward extensions of X can be safely eliminated. In this case, the number of pruned itemsets can reach up to $T(L, i) = O\left((N - i)^{n-L}\right)$. This implies that the earlier (i.e., for smaller values of L) and more frequently the pruning conditions hold, the greater the number of unpromising candidates that are eliminated. In essence, these pruning strategies significantly reduce the search space, which explains the strong performance of both MaxFHUOI-Miner and CGFHUOI-Miner compared to existing state-of-the-art algorithms.

V. EXPERIMENTAL EVALUATION

This section presents a detailed performance analysis of the proposed algorithms in comparison to both baseline and previously established methods for mining concise representations of FHUOIs. The experiments were conducted on a system running Windows 10, equipped with an Intel(R) Core(TM) i5-5200U 2.2 GHz CPU and 12 GB of RAM. A key highlight of this study is that CGFHUOI-Miner and MaxFHUOI-Miner are the first algorithms designed to discover both CFHUOIs and GFHUOIs at the same process, as well as to mine only MFHUOIs, respectively. To ensure a comprehensive evaluation, MaxFHUOI-Miner was compared against the baseline algorithm BL-MaxFHUOI-Miner, which first extracts the complete set of FHUOIs from the dataset using SHO [6], and then identifies all MFHUOIs in a post-processing step. Furthermore, the performance of CGFHUOI-Miner was assessed against CloFHUOIM [7] and Gen-FHUOIM [8][9], denoted as CloFHUOIM+Gen-FHUOIM, focusing on runtime efficiency and memory consumption. Notably, CloFHUOIM was developed specifically for extracting CFHUOIs, while Gen-FHUOIM was designed for mining GFHUOIs. To ensure a fair and meaningful comparison, the runtime of CGFHUOI-Miner was benchmarked against the combined runtime of CloFHUOIM and Gen-FHUOIM for each tested min_sup and min_uo value pair. Likewise, the maximum memory usage was determined by evaluating the peak memory consumption of CloFHUOIM and Gen-FHUOIM.

All algorithms were implemented in Java 8 SE to ensure a consistent and reliable platform for performance evaluation. The experiments were conducted using synthetic datasets generated by the IBM Quest data generator (as referenced in [30]), which provides extensive control over key parameters, including average transaction length(T), average size of maximal frequent itemsets (I), number of distinct items (N), and dataset size (D) (measured in thousands of transactions). To comprehensively assess algorithm performance, evaluations were carried out on both real-world QDBs, Connect, Chess, Mushroom, and BMS, and synthetic QTDBs T20I4N600D- and C20D10K. The real-world QTDBs and C20D10K were obtained from [30], while T20I4N600D- was generated using the IBM

Quest data generator. The tested QTDBs exhibit a range of density levels: Connect and Chess are highly dense, Mushroom and C20D10K are moderately dense, while BMS and T20I4N600D- are sparse. Table 5 summarizes their key characteristics, including the number of transactions, number of distinct items, average transaction length, density, and data type.

Table V
CHARACTERISTICS OF THE DATASETS

Dataset	#Trans. (D)	#Items (N)	Average trans. length (T)	Type of data
Chess	3,196	76	37	Real-life
Connect	67,557	129	43	Real-life
Mushroom	8,124	119	23	Real-life
c20d10K	10,000	192	20	Synthetic
T20I4N600D-	100K-160K	600	20	Synthetic
BMS	59,602	497	2.51	Real-life

1. Performance evaluation of the proposed algorithms

This subsection evaluates and compares the performance of the proposed MaxFHUOI-Miner and CGFHUOI-Miner algorithms against the baseline and previous approaches, focusing on runtime and memory usage across four datasets: Mushroom, Chess, BMS, and C20D10K.

a) Performance of the MaxFHUOI-Miner algorithm

The results in Figure 11 show that the runtime of the MaxFHUOI-Miner algorithm decreases as *min_sup* and *min_uo* increase. This trend is expected, as higher values of *min_sup* and *min_uo* significantly reduce the size of the discovered result sets (as shown in Figure 13).

Notably, MaxFHUOI-Miner outperforms the baseline algorithm by a factor of 10 to 21 times (in runtime) on three tested datasets, Mushroom, BMS, and C20D10K on average, particularly when *min_sup* and *min_uo* are set to lower values. On the Chess dataset, it runs approximately three times faster than BL-MaxFHUOI-Miner across all tested values of *min_sup* and *min_uo*. This significant speedup is due to the efficient pruning strategies employed by MaxFHUOI-Miner, which enable the early elimination of many unpromising candidates that cannot be MFHUOIs. By effectively reducing the search space, the algorithm not only accelerates the mining process but also minimizes memory consumption.

In contrast, BL-MaxFHUOI-Miner takes a less efficient approach: it first generates a large set of all FHUOIs and then examines each itemset individually to determine whether it qualifies as an MFHUOI. If an itemset does not meet the criteria, only a single node/itemset is removed at a time. This sequential pruning process is computationally intensive, leading to significantly higher time and memory consumption.

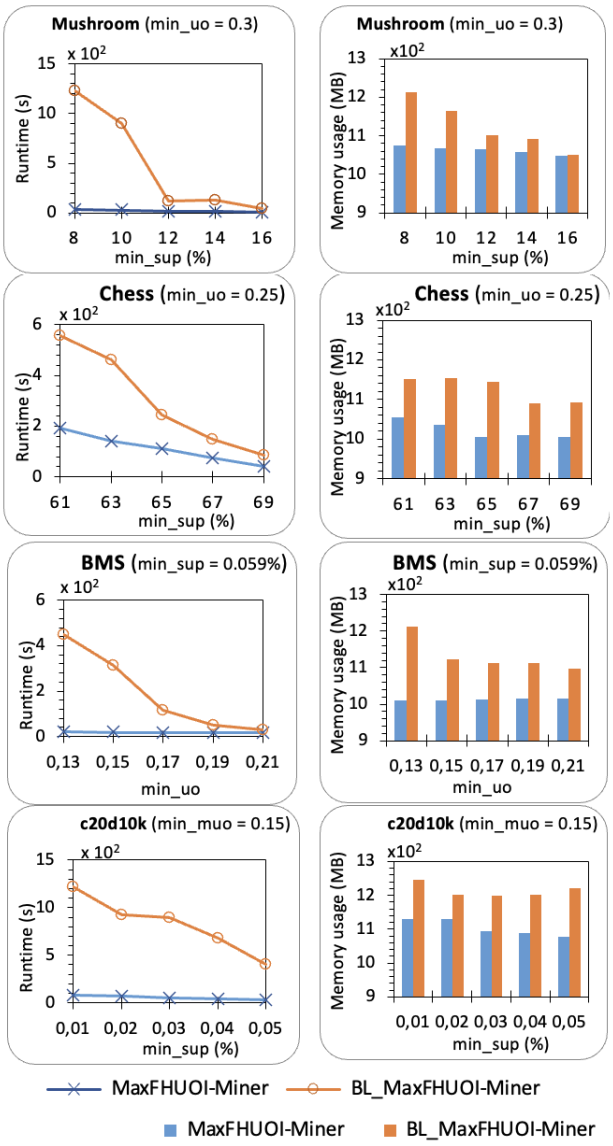


Figure 11. The performance of the MaxFHUOI-Miner algorithm.

b) Performance of the CGFHUOI-Miner algorithm

The results in Figure 12 clearly demonstrate the efficiency and effectiveness of the CGFHUOI-Miner algorithm in simultaneously discovering both CFHUOIs and GFHUOIs. By leveraging the novel EPNonCGBr and LP-NonCGBr pruning strategies, introduced in Corollary 1 and Corollary 2, the algorithm efficiently filters out non-closed and non-generator FHUOIs at an early stage, leading to significant reduction in both runtime and memory consumption across all datasets.

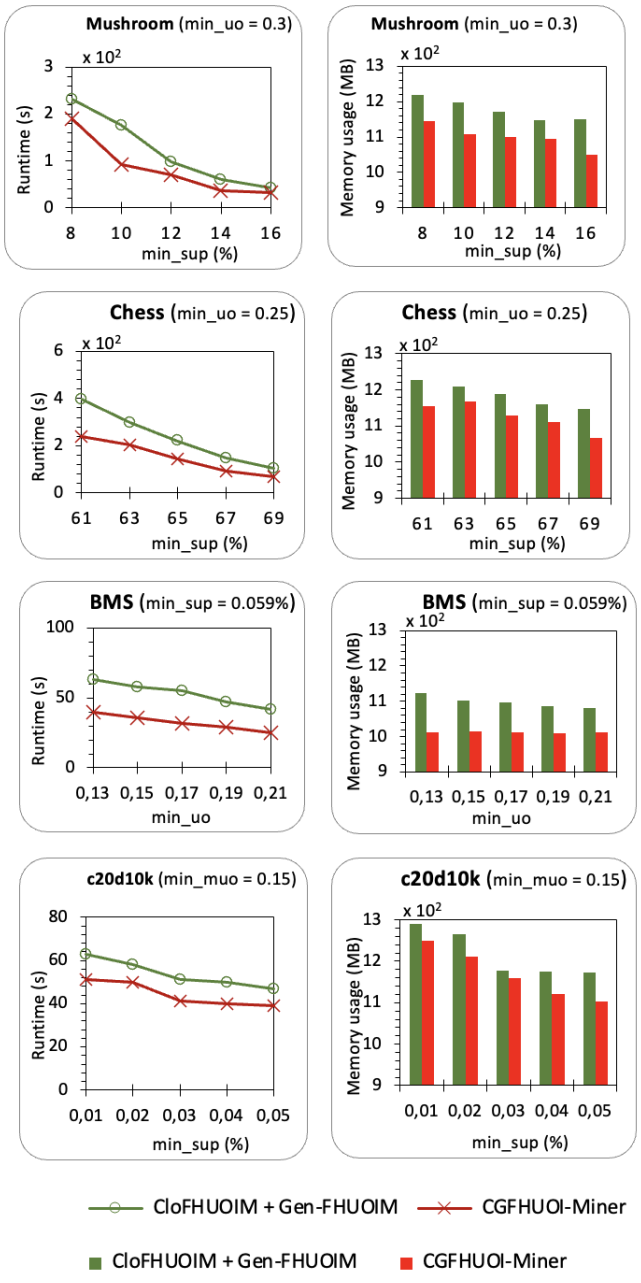


Figure 12. The performance of the CGFHUOI-Miner algorithm.

Unlike the previous CloFHUOIM + GenFHUOIM approach, which separately extracts CFHUOIs (the CloFHUOIM algorithm) and GFHUOIs (the Gen-FHUOIM algorithm), CGFHUOI-Miner performs both tasks in a single process, resulting in substantial computational savings. The notable improvements are observed in datasets such as Mushroom and BMS, where redundant patterns significantly increase computational costs.

Additionally, the proposed algorithm also stays efficient across various *min_sup* and *min_uo* thresholds, showing its strength in different mining conditions.

2. Comparison of result sets from the proposed and previous algorithms

As noted in the Introduction, the concise representation sets, $\mathcal{CFHUOI}$, $\mathcal{GFHUOI}$ and $\mathcal{MFHUOI}$, are significantly smaller than the complete $\mathcal{FHUOI}$ set, which contains all FHUOIs. These compact representations enable faster mining, reduced memory usage, and improved interpretability. To demonstrate their effectiveness on both real and synthetic datasets, Figure 13 compares the number of CFHUOIs, GFHUOIs, and MFHUOIs with the total number of all FHUOIs, mined using SHO [6], a state-of-the-art algorithm for extracting all FHUOIs. To verify correctness, the CFHUOIs and GFHUOIs generated by CGFHUOI-Miner are compared with the outputs of CloFHUOIM [7] and Gen-FHUOIM [8][9], respectively. Likewise, MFHUOIs produced by MaxFHUOI-Miner are validated against those obtained from BL_MaxFHUOI-Miner, which filters MFHUOIs from the output of the SHO algorithm. The identical results confirm the correctness and completeness of the proposed algorithms.

As shown in Figure 13, the numbers of CFHUOIs, GFHUOIs, and MFHUOIs are consistently much smaller than the number of all FHUOIs across the four tested datasets, Mushroom, Chess, BMS, and c20d10k, under various values of *min_sup* and *min_uo*. For instance, on c20d10k, GFHUOIs, CFHUOIs, and MFHUOIs account for only 4.63%, 1.65%, and 0.2%, respectively, of all FHUOIs. Notably, $\mathcal{MFHUOI}$ is the most compact, representing just 4.2% and 11.8% of the sizes of $\mathcal{CFHUOI}$ and $\mathcal{GFHUOI}$. These results emphasize the efficiency and practicality of mining concise representations over extracting the complete $\mathcal{FHUOI}$ set.

3. Scalability

This section analyzes the scalability of the compared algorithms by modifying the parameters of the synthetic dataset, as outlined in Table 5, along with the database sizes. In Figure 14 and Figure 15, a '-' symbol is added to the dataset name to indicate changes in the preceding parameter. The synthetic datasets for these different parameter settings were generated using the IBM Quest Synthetic Data Generator. The figures illustrate a comparative evaluation of the scalability of MaxFHUOI-Miner and CGFHUOI-Miner in terms of runtime and memory consumption on two large QTDBs from Table 5, Connect and T20I4N600D-. To preserve the density characteristics of each test database, the values of *min_sup* and *min_uo* thresholds remain unchanged

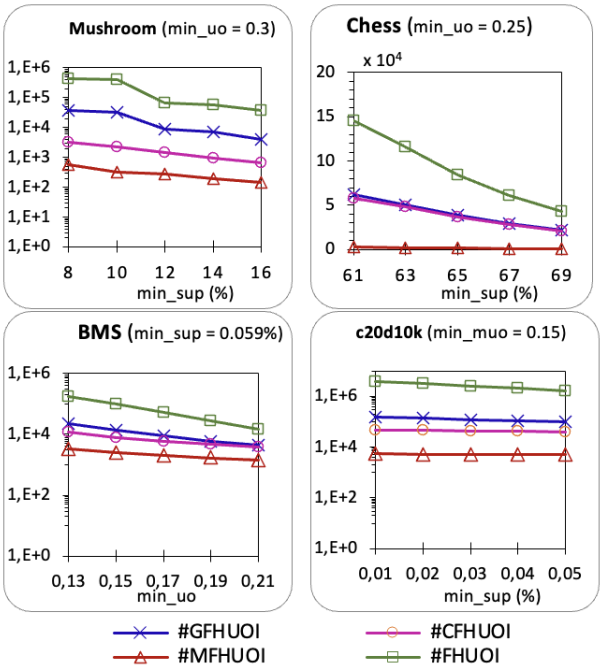


Figure 13. The number of FHUOIs and concise representations.

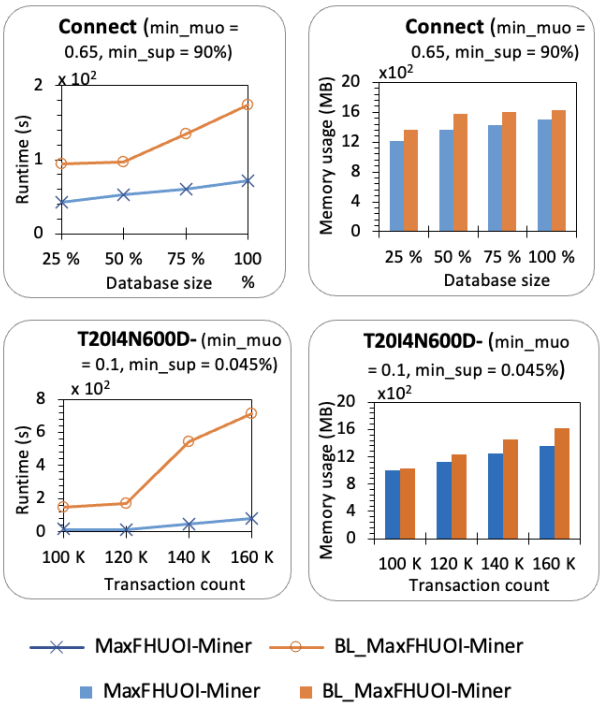


Figure 14. The scalability of the MaxFHUOI-Miner algorithm.

while the database size varies. Specifically, the size of the Connect dataset is adjusted to 25%, 50%, 75%, and 100%, while the transaction count of T20I4N600D- is modified from 100K to 120K, 140K, and 160K. In terms of runtime, the proposed algorithms demonstrate superior scalability as dataset size and transaction count increase, whereas the runtime of baseline and previous algorithms rises at a significantly faster rate. A comprehensive evaluation shows that the proposed algorithms are faster than existing methods and use significantly less memory. The rationale behind this observation aligns with the explanation provided in Subsection 5.1.

Overall, the results strongly validate the practicality and scalability of the proposed algorithms CGFHUOI-Miner and MaxFHUOI-Miner in mining concise representations of FHUOIs. The integration of advanced pruning strategies not only enhances computational efficiency but also optimizes memory utilization, making CGFHUOI-Miner and MaxFHUOI-Miner highly efficient solutions for discovering these representations. These advantages position the algorithms as state-of-the-art approaches for discovering informative and compact patterns of FHUOIs, making it promising methods for real-world high-utility occupancy itemset mining applications.

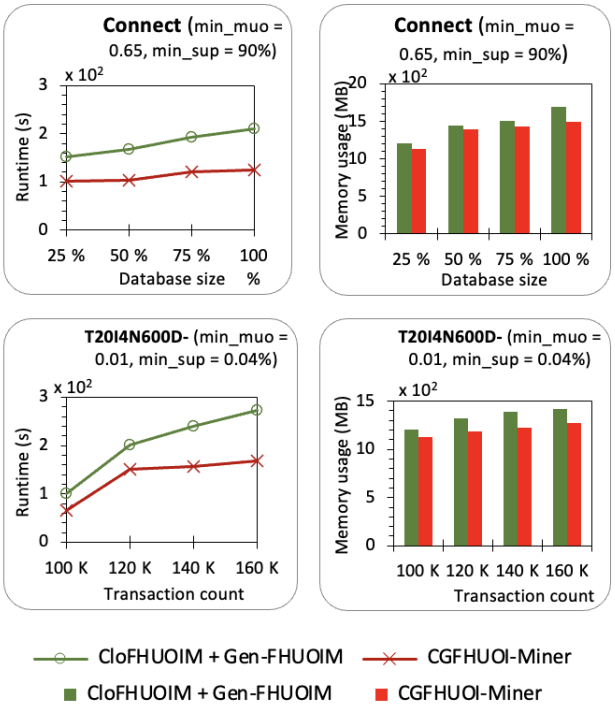


Figure 15. The scalability of the CGFHUOI-Miner algorithm.

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces two novel algorithms: MaxFHUOI-Miner, for discovering only MFHUOIs, and CGFHUOI-Miner, for simultaneously mining both CFHUOIs and GFHUOIs in a single process. To efficiently prune branches containing non-closed, non-generator, or non-maximal FHUOIs in the prefix tree, these algorithms incorporate novel and extended pruning strategies. The LPNonMaxBr strategy operates at two consecutive levels in the prefix search tree, enabling the early elimination of non-maximal FHUOIs, while the LPNonCGBr strategy leverages information from three successive levels to quickly discard branches that cannot contain both CFHUOIs and GFHUOIs. Notably, these strategies eliminate the need to verify subset relationships among itemsets, significantly reducing computational costs. Additionally, the EPNonMaxBr and EPNonCGBr strategies are introduced to facilitate the early elimination of redundant patterns through backward checking. Experimental evaluations demonstrate that the proposed algorithms outperform baseline and existing methods in terms of runtime, memory efficiency, and scalability.

Looking ahead, we plan to develop algorithms for extracting concise representations of frequent high average-utility and utility occupancy sequences in quantitative sequence databases.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2021.52.

REFERENCES

[1] Z. H. Deng, "Mining high occupancy itemsets," *Future Generation Computer Systems*, vol. 102, pp. 222–229, 2020.

[2] K. Zhang, Y. Zhang, and Z. Wang, "Frequent pattern mining based on occupation and correlation," in *Proc. IEEE 3rd Int. Conf. Electronic Information and Communication Technology (ICEICT)*, pp. 161–166, 2020.

[3] H. Kim *et al.*, "Mining high occupancy patterns to analyze incremental data in intelligent systems," *ISA Transactions*, vol. 131, pp. 460–475, 2022.

[4] L. T. T. Nguyen, T. Mai, G. H. Pham, U. Yun, and B. Vo, "An efficient method for mining high occupancy itemsets based on equivalence class and early pruning," *Knowledge-Based Systems*, vol. 267, Art. no. 110441, 2023.

[5] B. Shen, Z. Wen, Y. Zhao, D. Zhou, and W. Zheng, "OCEAN: Fast discovery of high utility occupancy itemsets," in *Lecture Notes in Computer Science*, pp. 354–365, 2016.

[6] J. He, X. Han, J. Wang, and K. Zhang, "Efficient high-utility occupancy itemset mining algorithm on massive data," *Expert Systems with Applications*, vol. 210, Art. no. 118329, 2022.

[7] H. Duong, H. Pham, T. Truong, and P. Fournier-Viger, "Efficient algorithms to mine concise representations of frequent high utility occupancy patterns," *Applied Intelligence*, 2024.

[8] D. Hai and T. Tin, "An efficient algorithm for mining frequent high utility occupancy generators," in *Proc. 9th Int. Conf. Cloud Computing and Internet of Things (CCIOT)*, pp. 101–109, 2025.

[9] D. Hai, H. Tien, and T. Tin, "Mining concise representations of frequent high utility occupancy itemsets using generator patterns," *Dalat University Journal of Science*, to be published, 2025.

[10] J. Li, H. Li, L. Wong, J. Pei, and G. Dong, "Minimum description length principle: Generators are preferable to closed patterns," in *Proc. 21st AAAI Conf. Artificial Intelligence*, pp. 409–414, 2006.

[11] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proc. SIAM Int. Conf. Data Mining*, pp. 482–486, 2004.

[12] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in *Proc. Int. Workshop Utility-Based Data Mining*, pp. 90–99, 2005.

[13] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proc. ACM Int. Conf. Information and Knowledge Management*, pp. 55–64, 2012.

[14] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 253–262, 2010.

[15] S. Zida *et al.*, "EFIM: A highly efficient algorithm for high-utility itemset mining," in *Proc. Mexican Int. Conf. Artificial Intelligence (MICAI)*, pp. 530–546, 2015.

[16] S. Krishnamoorthy, "HMiner: Efficiently mining high utility itemsets," *Expert Systems with Applications*, vol. 90, pp. 168–183, 2017.

[17] J.-F. Qu *et al.*, "Mining high utility itemsets using prefix trees and utility vectors," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 10224–10236, 2023.

[18] V. S. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient algorithms for mining the concise and lossless representation of high utility itemsets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 726–739, 2015.

[19] W. Gan *et al.*, "HUOPM: High-utility occupancy pattern mining," *IEEE Transactions on Cybernetics*, vol. 50, pp. 1195–1208, 2020.

[20] C. M. Chen *et al.*, "Discovering high utility-occupancy patterns from uncertain data," *Information Sciences*, vol. 546, pp. 1208–1229, 2021.

[21] S. Vemulapalli and S. Mogalla, "High utility-occupancy sequential pattern mining algorithm based on utility-occupancy framework," *Int. J. Engineering Trends and Technology*, vol. 69, pp. 228–235, 2021.

[22] C.-W. Wu, P. Fournier-Viger, J. Gu, and V. S. Tseng, "Mining compact high utility itemsets without candidate generation," in *High-Utility Pattern Mining*, Studies in Big Data, pp. 283–307, 2019.

[23] H. Duong *et al.*, "Efficient algorithms for mining closed and maximal high utility itemsets," *Knowledge-Based Systems*, vol. 257, Art. no. 109921, 2022.

[24] C.-W. Wu, P. Fournier-Viger, J.-Y. Gu, and V. S. Tseng, "Mining closed high utility itemsets without candidate generation," in *Proc. Conf. Technologies and Applications of Artificial Intelligence*, pp. 187–194, 2015.

[25] P. Fournier-Viger *et al.*, "EFIM-Closed: Fast and memory-efficient discovery of closed high-utility itemsets," in *Proc. Int. Conf. Machine Learning and Data Mining in Pattern Recognition*, pp. 199–213, 2016.

[26] L. T. T. Nguyen *et al.*, "An efficient method for mining high utility closed itemsets," *Information Sciences*, vol. 495, pp. 78–99, 2019.

[27] P. Fournier-Viger, C.-W. Wu, and V. S. Tseng, "Novel concise representations of high utility itemsets using generator

patterns," in *Proc. Int. Conf. Advanced Data Mining and Applications*, pp. 30–43, 2014.
[28] C.-W. Lin, T.-P. Hong, and W.-H. Lu, "An effective tree structure for mining high utility itemsets," *Expert Systems with Applications*, vol. 38, 2011.
[29] H. Duong, T. Truong, B. Le, and P. Fournier-Viger, "CG-FHAUI: An efficient algorithm for mining succinct pattern sets of frequent high average utility itemsets," *Knowledge and Information Systems*, 2024.
[30] P. Fournier-Viger *et al.*, "SPMF: A Java open-source pattern mining library," *Journal of Machine Learning Research*, vol. 15, pp. 3569–3573, 2014.

Tien Hoang received his MSc degree in Computer Science from the University of Science, VNUHCMC, Vietnam, in 2009. He is currently a lecturer in the Department of Mathematics and Computer Science at Dalat University, Vietnam, and a research student at the University of Science, Ho Chi Minh City. His research focuses on artificial intelligence and data mining.
Email: tienhoang@dlu.edu.vn

Lan Huynh received her Master's degree in Computer Science from the University of Science, VNU-HCMC, in 2011. She is currently a lecturer in the Faculty of Information Technology at Ho Chi Minh City University of Industry and Trade, Vietnam. Her research focuses on artificial intelligence and data mining.
Email: lanhuynh@dlu.edu.vn

Hai Duong is an Associate Professor at Dalat University, Vietnam. He obtained his Ph.D. in Computer Science in 2020 from the University of Science, VNU-HCMC. During his doctoral studies, he received several awards for outstanding scientific research. Currently, he serves as both a lecturer and researcher in the Faculty of Mathematics and Computer Science at Dalat University, where he also holds the position of Vice Head. His primary research interests include artificial intelligence and data mining.
Email: haidv@dlu.edu.vn

Tin Truong is a researcher at the Department of Mathematics and Computer Science, University of Dalat, Vietnam. He received his B.S. degree in Mathematics from Dalat University in 1983 and his Ph.D. in Stochastic Optimal Control in 1990 at the Vietnam National University, VNU-Hanoi, Vietnam. His current research interests are artificial intelligence and data mining.
Email: tintruong@dlu.edu.vn

Distillation-Centric Approaches in Visual Question Answering with Mixture of Experts

Huy Hoang Huynh[1,2], Tung Le[1,2], Huy Tien Nguyen[1,2,*]

[1]Faculty of Information Technology, University of Science, Ho Chi Minh city, Vietnam
[2]Vietnam National University, Ho Chi Minh city, Vietnam

Correspondence: Huy Tien Nguyen. Email: ntienhuy@fit.hcmus.edu.vn
Communication: received 30/11/2024, revised 11/03/2025, accepted 04/05/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1352

***Abstract*: Recent advancements in computer vision and natural language processing were applied to the Visual Question Answering task. Nonetheless, a significant proportion of models exhibiting high accuracy possess extensive architectural components. This has a significant impact on the process of bringing the technology to practical applications such as assistive devices for the blind and visually impaired, and other related fields. Our research focuses on compressing the Visual Question Answering model on the Vietnamese dataset by utilizing the knowledge distillation method. Furthermore, in order to enhance precision, we have also developed a Mixture of ViVQA Experts system that will adapt to each type of question for improving accuracy while increasing only a few parameters and not wasting time retraining the entire system from scratch. With a total of 204M parameters, this approach has reduced the size by 24.51% compared to the original model while only reducing accuracy by 6.59% on the overall test set. More specifically, we have made accuracy improvements on each question type: "number" increased by 1.35% and "color" increased by 0.48% compared to our distillation model. The code and pretrained models are available at: https://github.com/huynhhoanghuy/Distillation_ViVQA.**

Keywords: *visual question answering, knowledge distillation, mixture of expert, vision-language.*

I. Introduction

With significant progress in vision tasks and textual tasks, recent research groups have focused more on problems that combine both fields, such as Visual Question Answering (VQA) [1], Visual Captioning [2], Visual Reasoning [3], etc. To take advantage of the techniques applied in each of these fields to vision-language tasks, most existing methods require significant adjustments. Among the tasks listed, VQA is a more challenging task, requiring not only a deep understanding of images and text but also finding the correlation between them to generate answers.

Visual question answering systems aim to answer text-based questions about images [4]. The key components of modern VQA systems include image feature extractors, text/question encoders, and classifier models that predict answers based on the encoded representations. While earlier work focused heavily on recurrent models and CNN-RNN architectures, recent progress has seen the increasing adoption of attention mechanisms and transformer-based approaches. For instance, Chao Yang *et al.* [5] proposed a dual-branch model with contextual constraints to limit bias. Hangbo Bao *et al.* [6] explored unified multimodal transformer encoders for VQA. Several works have also leveraged pretrained models, such as BEiT and BARTPho, to effectively encode visual and textual semantics.

Recent studies have presented substantial advancements in the development of large-scale models [5–10] which demonstrate notable improvements in VQA performance. As the VQA task requires a wide range of knowledge (e.g., position, color, counting etc), VQA's models necessitate a significant number of parameters to effectively encapsulate these abilities. Consequently, this requirement for extensive parameterization makes traditional model distillation approaches less effective in reducing the size of these large models without compromising their performance. In addition, the heterogeneity of question types within VQA, each necessitating distinct evaluation metrics (e.g., categorical cross entropy for object detection and mean squared error for counting tasks), makes the use of a single loss function sub-optimal in the training of VQA models. To address these challenges, we propose the utilization of a distilled model as a foundation framework for the development of a **M**ixture **o**f **V**iVQA **E**xperts (MoVE). In this work, we define the teacher model as a large-scale VQA model built with complex architectures (such as ViT and ResNet [7]) that are capable of achieving high accuracy across various types of questions. However, these models contain hundreds of millions of parameters, leading to slow inference and making deployment on resource-limited devices impracti-

cal. To address this challenge, we constructed a student model, which is a more compact version of the teacher model. The student model is trained using knowledge distillation [11–13], allowing it to inherit knowledge from the teacher by minimizing divergence between their outputs and intermediate features. This technique enables the student model to preserve much of the teacher's accuracy while significantly reducing the number of parameters and computation cost. We use this distilled student model as the foundation for developing a more efficient and adaptable architecture. This system is designed to specifically accommodate the nuanced requirements of various types of questions. According to experimental results, the accuracy of the entire system has improved by 54.37% compared to 54.17% in the student model. Here, accuracy is calculated as the ratio of the number of correctly predicted answers to the total number of questions in the test set. Additionally, the accuracy on color-related questions increased by 0.48%, and the accuracy on counting-related questions increased by 1.35%. The contributions of this work are summarized as follows:

- We introduce a novel knowledge distillation methodology which entails substituting the image feature extraction modules, specifically ViT and ResNet, with smaller variants. This strategy achieves a commendable accuracy of 54.17%, closely approaching the 60.76% accuracy of the original model while concurrently achieving 24.51% reduction in the number of parameters.
- We propose a mixture network strategy that promotes a unified architecture by merging fusion features derived from both image and question inputs. At the mixture network's top layers, we establish three classifiers catering to three distinct question types: color, number, and others. This innovation marginally enhances the system's performance and serves as a preliminary validation for the efficacy of segmenting a universal classifier into multiple specialized classifiers aligned with specific question domains.
- We present detailed comparisons and related experiments on downstream tasks, including specific question types. Experimental results show that the proposed approach significantly improves accuracy.

The structure of this paper is organized as follows: Section II presents some related research, focusing on the VQA problem, knowledge distillation, and a mixture of expert techniques. Section III delves into our methodology, describing how we distill a VQA model and apply a mixture of experts to it. Section IV provides an overview of our experiments and an analysis of our findings. Finally, Section VI summarizes our study and suggests directions for future.

II. Related Works

Recently, most VQA frameworks only focus on English, lacking data sets for other languages, especially Vietnamese. Besides, the Vietnamese Visual Question Answering (ViVQA) dataset [14] introduced by Khanh *et al.* is translated from the VQA dataset that has been used by several groups. This dataset containing 10,328 images from the MSCOCO dataset and 15,000 pairs of questions and answers in Vietnamese.

VQA model: The initial model [15] for this dataset employed recurrent neural networks and LSTM using GloVe word embeddings to compute image and question representations separately. These were forward to some textual/visual guided attention layers. After that, they are fused using point-wise addition and fed to a dense layer to projected into a vector for prediction. Later, [7] proposed a new approach, which using PhoBert for getting textual features and using both Vision Transformer (ViT) with ResNet for getting global and local visual features. The steps to combine them, are the same as the article mentioned before. Recently, there was a research group using BARTPho for text and BEIT for image [8]. Then, both types of features are concated together and passed through common Multi-head Self-Attention layers. This is the mechanism that allows the model to learn the correlation between visual and textual features. Finally, the model will predict the answer based on the fusion features that have been fed through the Feedforward Neural Network (FNN).

Model compression: While delivering superior results, modern deep neural networks present deployment challenges for resource-constrained platforms for deployment on devices such as mobile, jetson device, etc. due to their sheer complexity and massive computational demands. The larger the model, the longer the inference time increases, leading to a worse user experience. On the other hand, we see that recently, the field of model compression has gradually become popular, such as pruning, quantization, knowledge distillation, and so on. With pruning, pruning branches that have little influence in the model will help the compression ratio be 10 to 15 times, but there is a major weakness that is that with models with little sparseness, it is quite difficult to prune well [11]. With quantization, converting data types to a less computational and storage form and grouping calculations into clusters helps speed up execution and reduce the size of the file. However, this method still has problems when being implemented with sparse models [11]. On the other hand, the knowledge distillation method does not encounter that limitation and also has no limitation on compression ratio. In this approach, we begin with a teacher model, which is a large, high-capacity model designed with deep architectures

such as ResNet and ViT layers to handle complex tasks with strong accuracy. The teacher model typically contains hundreds of millions of parameters and is capable of learning rich multi-modal relationships. In contrast, the student model is a smaller, more lightweight version, designed with fewer parameters and simpler modules - for example, replacing large backbone networks with smaller ones like MobileNetV3. Despite its compact architecture, the student model is trained to learn from the teacher's knowledge. It simply creates a new student model which is smaller and more compact than the original teacher model [11]. Then, we use constraints between features at different levels, corresponding to each pair of layers, to transfer the knowledge that the teacher model has learned into the student model. Such as Humair Raj Khan *et al.* [12] proposed a method of knowledge distillation from the teacher monolingual VQA model to the student multilingual VQA model. They have used multiple objectives to transfer the knowledge: token embedding distillation; object attention distillation; prediction distillation. Jianfeng Wang *et al.* [13] also distill knowledge from the output and the intermediate layers on medical VQA datasets.

Knowledge Distillation: has been applied to the task of model compression. By using "knowledge" transfer technical from teacher model to student model, with the expectation that only a small student model can have the same knowledge as the large teacher model after going through the layers. Specifically, with a given sample (x_i, y_i), there is knowledge $f^T(x_i)$ in the teacher network (T) and knowledge $f^S(x_i)$ in the student network (S). In addition, the student model takes an input x_i and produces an output representation $S(x_i)$, which is trained to approximate the output of the teacher model $T(x_i)$ through the knowledge distillation process. To transfer knowledge, we should minimize the divergence between them:

$$Loss = L_S(S(x_i), y_i) + L_{KD}(f^T(x_i), f^S(x_i)) \quad (1)$$

, where $L_S(.)$ refers to the original supervised loss on the Student and label of sample. $L_{KD}(.)$ refers to a constraint that the knowledge of teacher and student should be the same. In particular, L_{KD} can be cross-entropy (CE), mean squared error (MSE), mean absolute error (MAE) function or KLdivergence (KL), ... For example, DistillBERT [16] proposed to train the small BERT model by imitating the teacher model's output probability distribution for the masked language prediction task, as well as mimicking the embedding features from the teacher. MobileBERT [17] and TinyBERT [18] distilled by utilizing the layer-wise attention distributions for comparing to the teacher model's attention distributions using MSE loss like as L_{KD}, and so on.

Mixture of experts: (MoE) [19] is an adaptive neural network architecture that allows combining multiple models (experts) to solve a problem. Each expert specializes in a certain area of the input space. Its architecture includes expert networks, a gating network to decide the combined weights of experts based on input, and a module that combines expert results according to weights [20, 21]. The strength of this model is its flexibility, which can be expanded by adding new experts without affecting old experts. MoE has been successfully applied in many fields, such as computer vision, natural language processing, speech recognition, etc. More specifically, there are a few studies on the VQA problem that have used MoE. For example, V Pahuja, et al. [22] applied MoE to optimize the CNN network using conditional computation, only partially activating the modules based on the question representation vector; G Liu, et al. [23] has proposed combining self-attention and cross-attention in cross-modal mixture experts (CMMEs) models, including visual experts and textual experts, to increase the interactivity of visual and linguistic features. In this study, we do not use MoE to optimize calculations or to increase the association between two types of features. We realize that for each type of question, there will be a separate domain. Therefore, it is necessary to have separate classifiers and a controller for navigation.

III. METHODOLOGY

In this section, we will first introduce our Mixture of ViVQA Experts system (MoVE). Given an image I and a question Q as input, this system will try to give the correct answer by classifying a list of candidate answers. The overview of our system is illustrated in Figure 1, the model consists of 3 main modules: the visual-textual representation extractor; the gating module for navigating input into expert; the classifier experts. This study will present a new approach for optimizing the Vietnamese VQA model on the Vietnamese Visual Question Answering dataset.

For the distillation phase, the NDA model [7] was designated as the teacher model, from which the distillation process commenced towards a more compact student model. This process entailed the substitution of two image extractor modules, namely ResNet and ViT, with their smaller counterparts. This strategic adjustment underscores our commitment to optimizing model's efficiency without substantially compromising its performance capabilities. This particular aspect will be elaborated upon in the specific context of Section III.1.

As mentioned above, teacher models employ architectures with many parameters to deal with various domains of questions, while small student models have the limited

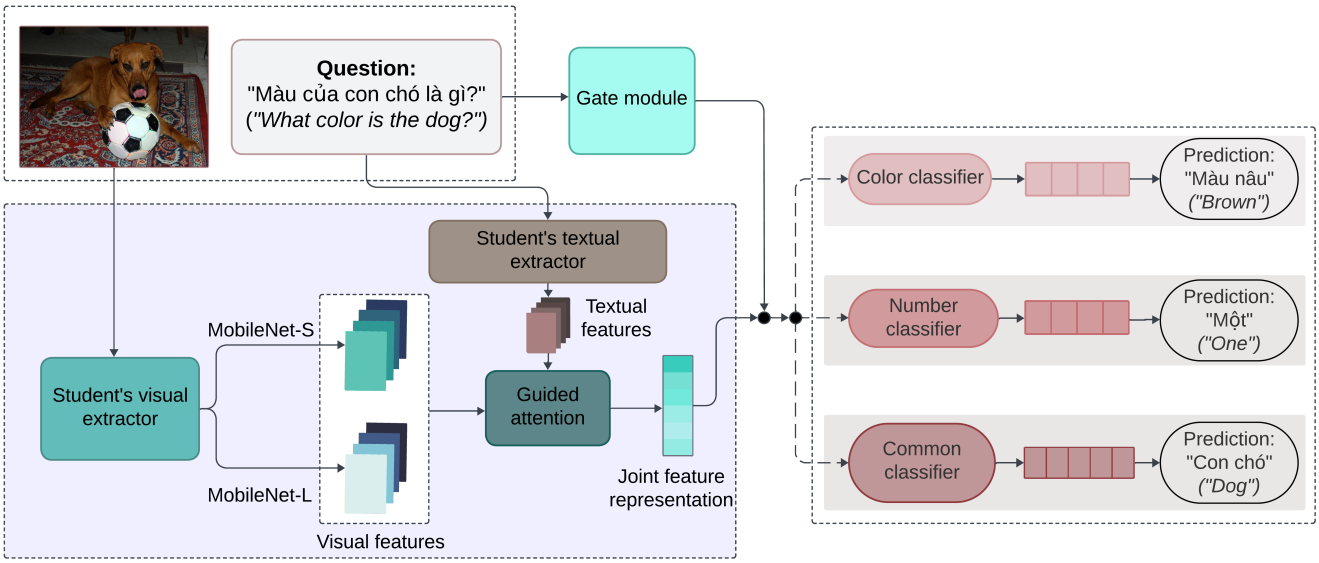


Figure 1. Our Mixture of ViVQA Experts system (MoVE). MobileNet-S and MobileNet-L stand for the MobileNetV3 model in small and large sizes, respectively. The content in parentheses is the English translation.

number of parameters. To overcome this challenge, we design separate top layers for modeling different types of tasks and mixing them together in one system to save parameters while maintaining good performance. In recent the Mixture-of-Expert paradigms [6, 19, 24], these methods all use a certain input signal to pass through a baseline model to predict which experts will receive the signal. To avoid increasing the number of parameters and computations by a large amount, we propose MoVE, an efficient rule-based pipeline for navigating inputs into corresponding experts. We analysed results and decided to build a mixture network with multiple classifiers corresponding to three specialized question types: *color*, *number*, and *others*. The *others* category includes the *object* and *location* questions. Section III.2 will provide an in-depth exploration of this particular detail.

1. Distill visual features

Given local visual features $f_{v_1} \in \mathbb{R}^{m_1 \times n_1}$, global visual features $f_{v_2} \in \mathbb{R}^{m_2 \times n_2}$, and textual features $f_t \in \mathbb{R}^{max\_len \times n_2}$, the original model [7] can combine f_{v_1} and f_{v_2} with f_t into fusion features $f_{v_1 \leftarrow t}, f_{v_2 \leftarrow t} \in \mathbb{R}^{1 \times n_2}$ by guided attention mechanism followed up textual features [7, 25] and forwarding to some reducing layers. Then the authors combined them into multi-context visual features $f_{joint\_visual} \in \mathbb{R}^{1 \times (n_1 + n_2)}$. They apply guided attention followed up f_{joint_visual} on f_t to get better textual features $f_{t \leftarrow v_j} \in \mathbb{R}^{1 \times n_2}$. Finally, they fusion $f_{t \leftarrow v_j}$ and f_{joint_visual} into joint features $f_{joint} \in \mathbb{R}^{1 \times d}$ and forward into classifier.

We assume there are other visual features of different smaller dimensions that can fuse with textual features. Our target is finding 2 smaller visual feature extraction modules and fuse that features with textual features without significant reduction in accuracy. On the other hand, we found a good candidate model: MobileNetV3 [26] which is both so light and is applied in parallel to the Resnet layers. Then we use 2 models: Large MobileNetV3 and Small MobileNetV3 for replacing ViT and ResNet-34 extractors (Fig. 2) and we call the new model is student model, which will learn knowledge from teacher model (the original model).

a) Problem formulation for distillation

Given an image I and a question Q as input, the target is finding the anwser A from a set of candidate anwsers O. The objective function is:

$$A = \arg\max_{A \in O} P(A|I, Q, O). \tag{2}$$

With traditional loss function for classification task, it only requires a constraint between prediction A and ground truth *label*:

$$Loss = L_S(A, label). \tag{3}$$

However, the student model needs additional knowledge from the teacher model to improve its precision. So that, the general loss function should be:

$$Loss = L_S(A, label) + L_{KD}(f^T(I, Q), f^S(I, Q)). \tag{4}$$

b) Our distillation method

With $L_S(.)$, we use cross-entropy loss (CE) as a traditional option.

$$L_S(A, label) = CE(A, label). \tag{5}$$

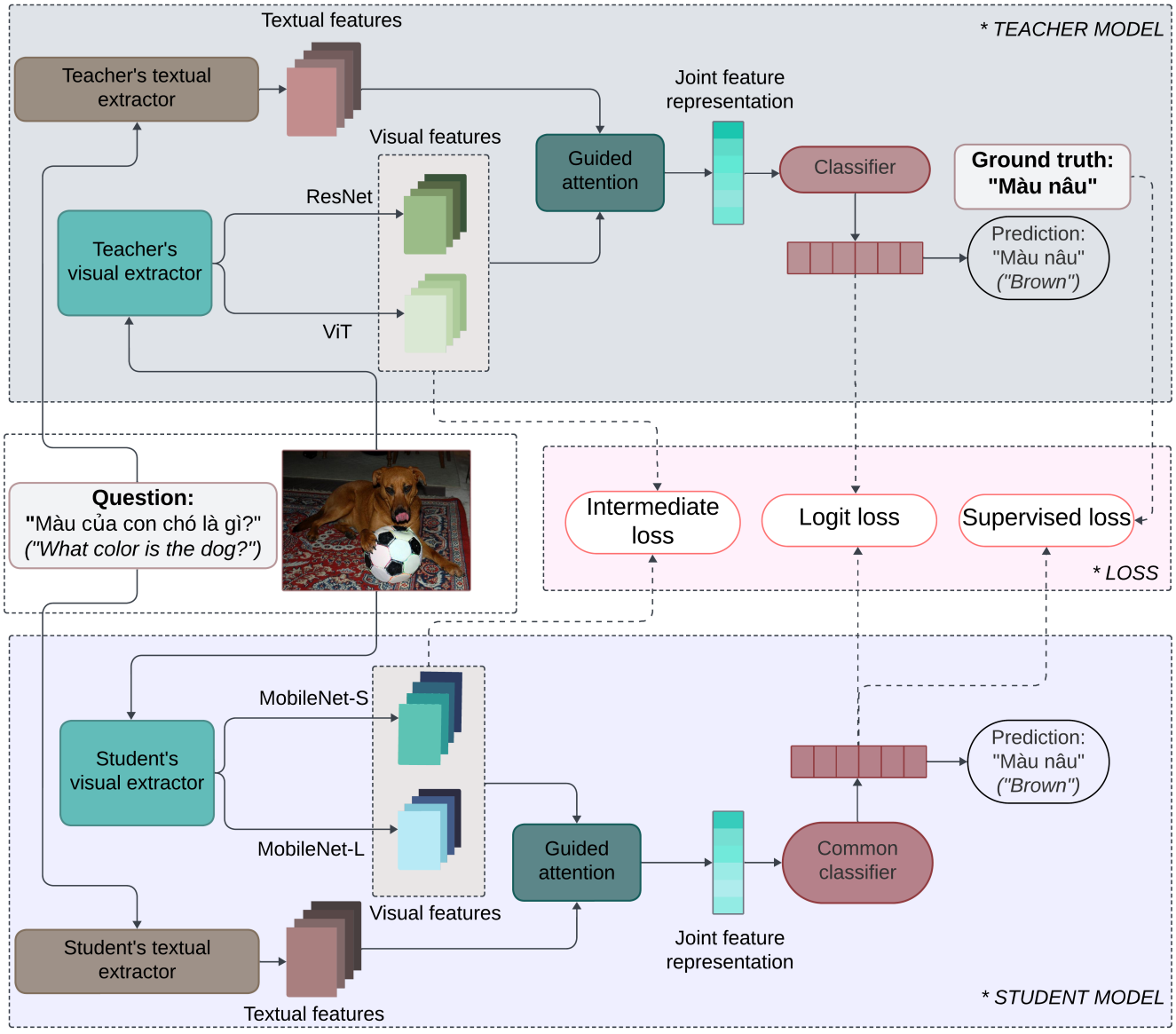


Figure 2. Illustration of the VQA model knowledge distillation flow. The content in parentheses is the English translation.

Beside that, our $L_{KD}(.)$ formulation is quite complex. We have tried to tie in student knowledge and teacher knowledge by making more constraints in intermediate features and last features (logit outputs). For intermediate features, the teacher model contains two layers: one before applying the attention mechanism and one after. In this study, we use the features at the beginning (before applying attention) to constrain the training process. Functionally, we use Kullback-Leibler divergence (KL) as the loss function for these two pairs of intermediate features. In detail:

$$\begin{aligned} L_{KD}(f^T(I,Q), f^S(I,Q)) &= L_{logit}(.) + L_{intermediate}(.) \\ &= CE(S(I,Q), T(I,Q)) \\ &+ KL(f^T_{v_1}, f^S_{v_1}) + KL(f^T_{v_2}, f^S_{v_2}) \end{aligned} \tag{6}$$

Each component of the loss function also has an associated weight factor. In addition, we are also using augmentation methods and temperature for the teacher's output and student's output to improve training progress. The formula 7 is our final loss function:

$$Loss = \alpha \cdot CE_{sup} + \beta \cdot \tau^2 \cdot CE_{logit} + \gamma \cdot KL_{f_1} + \delta \cdot KL_{f_2}. \tag{7}$$

where:

- CE_{sup}, CE_{logit}, KL_{f_1}, and KL_{f_2} is shortened form of $CE(A, label)$, $CE(S(I,Q), T(I,Q))$, $KL(f^T_{v_1}, f^S_{v_1})$, and $KL(f^T_{v_2}, f^S_{v_2})$.

- α, β, γ, and δ are the coefficients of CE_{sup}, CE_{logit}, KL_{f_1}, and KL_{f_2}.
- τ is temperature for distillation.

2. Adapting models to question categories through mixture of experts

In the dataset under investigation, we identify four distinct categories of questions: "what", "where", "number", and "color", each characterized by its own domain of values. Notably, the categories "number" and "color" exhibit constrained value ranges and possess inherent ordinal relationships. For instance, within the "number" category, there exists a sequential correlation such that the prediction of a specific value (e.g., "seven") implies elevated likelihood estimations for adjacent numerical classes and diminished probabilities for numerically distant classes. Conversely, while the "color" category also demonstrates ordinality, particularly when analyzed in the Hue, Saturation, Value (HSV) color space, it lacks a linear, countable progression.

Acknowledging these nuances, we advocate for a tailored approach in which distinct objective functions are designated for each question type. Consequently, our proposed system incorporates three specialized classifiers: a color classifier with ten classes, a number classifier also with ten classes, and a general classifier encompassing 353 classes. This methodology aims to optimize the accuracy and efficacy of the system by acknowledging and adapting to the specific characteristics of each question type.

We built a gate module, shown in Fig. 3, that acts as a bridge between input features and a classifier. This module receives a raw question as input. It will detect important keywords to identify what type of question the question is. For misleading keywords, we will consider a few other keywords to determine more precisely. In cases where there are no keywords, the question will be classified as a "what" question. In addition, we will train the classifier module's color and number for each data type while keeping the weights of the previous feature extractors intact because our system (MoVE) uses the same visual-textual feature representation extractor. This saves a large amount of computations for the system.

Regarding the question about "number", there is an ordering relationship between classes. For the "color" classifier training process, instead of applying cross entropy (CE) as a loss function, we expect squared Earth Mover's Distance (squared EMD) as mentioned by Le Hou et al. [27], will able to force classes that are far away from candidate classes to have lower prediction values than nearby classes. For the question about "color", we arrange the color classes by HSV, with H first, V next, and S last, in ascending order. To speed up the training process, we

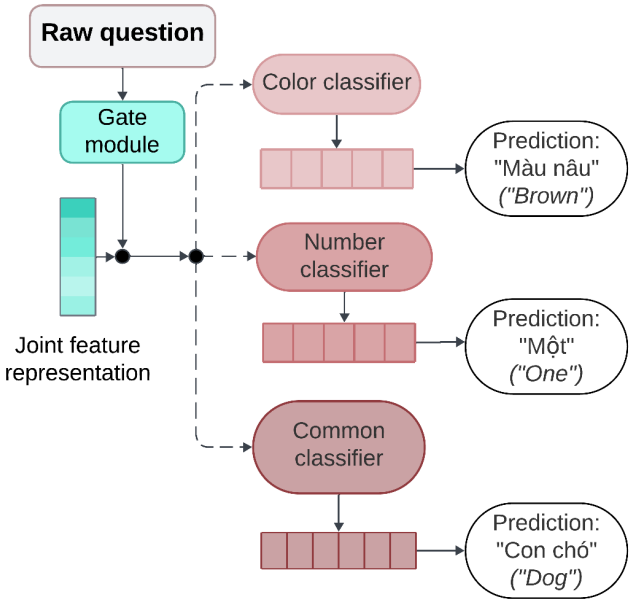


Figure 3. The application of a mixture of experts in the VQA system: Gate module with multiple classifier. The content in parentheses is the English translation.

use both squared EMD and CE because the relationships between color classes are less obvious than between number classes. Figure 1 illustrates our VQA system using a mixture of experts.

$$EMD^2 = \sum_{i=1}^{C} (CDF_i(p) - CDF_i(t))^2 \tag{8}$$

where:

- p and t are probability vector that sums to 1 of prediction and target.
- $CDF_i(.)$ is the i-th element of the $CDF(.)$ which is a function that returns the cumulative density function of its input.

In terms of architecture, we use a multi-layer perceptron (MLP) as specific classifier to encode the visual-textual features. Equation 9 and Equation 10 represent the color classifier and the number classifier, respectively.

$$\begin{aligned} h_1 &= dropout(tanh(x)) \\ h_2 &= dropout(tanh(W_2 h_1 + b_2)) \\ h_3 &= dropout(tanh(W_3 h_2 + b_3)) \\ p &= softmax(W_4 h_3 + b_4) \end{aligned} \tag{9}$$

where W_2, b_2, W_3, b_3, W_4, b_4 are the weight parameters. x is visual-textual features. p is the probability of the final answer.

$$h_1 = dropout(LeakyReLU(W_1 x + b_1))$$
$$p = softmax(W_2 h_1 + b_2) \quad (10)$$

where W_1, b_1, W_2, b_2 are the weight parameters. x is visual-textual features. p is the probability of the final answer.

IV. Experiments

In this section, first, we will demonstrate that our distillation method outperform on datasets with some strong constraints: intermediate features pairs, logit features pairs, and supervised loss between student output and label. Second, we will show our improvement in a mixture system constructed by that distilling model.

1. Experiment settings

We conduct experiments on Pytorch with the widely used Transformers library. All experiments are running on two GeForce RTX 3080 GPUs. The Gaussian noise (kernel size = (5, 9), sigma = (0.1, 5)) and random rotation (angle = 15°) are adopted as data augmentation. Regarding data preprocessing, we follow the same process as training the teacher model [7]. Table I shows the best hyperparameters we found. In addition, for knowledge distillation processing, we have decided to choose Eq. 11 is a final loss function :

$$Loss = 0.1 \cdot CE_{sup} + 0.3 \cdot 5^2 \cdot CE_{logit} + 0.3 \cdot KL_{f_1} + 0.3 \cdot KL_{f_2} \quad (11)$$

Table I
Hyperarameters settings.

Hyperparameters	Value
Temperature (τ)	5
Drop out	0.15
Learning rate	5e-4
Batch size	16

Furthermore, the number of parameters and the accuracy of the teacher model are 269,223,078 and 60.76%, respectively. This is considered a milestone to compare with the student model.

2. Experimental results

a) Distillation results

1) *Determine the feasibility of knowledge distillation:* Before focusing on this research, we need to check the feasibility of distillation. We assume that if the compact student model is capable of learning knowledge from the complex teacher model, then it is at least capable of learning knowledge when both models have the same architecture. Our loss function is constructed using CE_{sup} and KL_{logit}, with a learning rate of 10^{-4} and a dropout rate of 0.3. This configuration gave approximately the same results as the teacher achieved. Specifically, we achieve an accuracy of 59.66%, compared to 60.76% of the original model. The accuracy is calculated as the percentage of correctly predicted answers out of the total number of questions in the test set. This result has reinforced our confidence in the distillation pipeline and the performance of the network output so that we can continue to develop the following parts.
2) *Determine which module to distill:* We sequentially explored distillation processes for the ResNet branch alone and subsequently for both the ResNet and ViT branches. Based on the results Table II, the distillation of a singular branch resulted in a model size reduction by only 9.47%, and a decrease in the number of parameters by 7.5%. Conversely, the distillation of two branches significantly enhanced the model's compactness, achieving 25.71% reduction in size, and 24.51% decrease in parameter count. However, such substantial reductions in parameter volume were accompanied by a decline in model accuracy. As the relatively minor reductions of the single-branch distillation process did not meet our expectations for model efficiency improvements, we elected to proceed with the dual-branch distillation strategy. Concurrently, we initiated efforts to identify and implement measures aimed at mitigating the adverse impact on accuracy, thereby balancing the trade-off between model compactness and performance efficacy.

Table II
Experiments of distillation on 1 and both 2 visual branches of the teacher model. Green arrows (↑/↓) signify positive trends of the evaluation metrics, while red arrows (↑/↓) signify negative trends.

Distillation	Gain
ResNet → MobileNetv3 (small)	Size: 995.8 MB (9.47%↓) Num of params: 248,859,142 (7.5%↓) Best training acc: 98.3 % (2.18%↑) Best testing acc: 58.5 % (3.72%↓)
ResNet → MobileNetv3 (small), ViT → MobileNetv3 (large)	Size: 816.0 MB (25.82%↓) Num of params: 203,945,157 (24.51%↓) Best training acc: 98.3 % (2.18%↑) Best testing acc: 38.1 % (37.29%↓)

3) *Determine which knowledge constraints are feasible:* As mentioned, in the distillation process, in addition to two objective function: the alignments of the student model's predictions with the teacher model's predictions and with the true labels.Our investigation

reveals that the incorporation of additional constraints at intermediate stages of the model could further enhance the efficacy of the distillation process. In the original model, there are two main intermediate features: before applying the guided attention method and after applying the guided attention method. According to Table III, utilizing features prior to the application of attention mechanisms offers substantial benefits within the context of model distillation.

Table III
EXPERIMENTS OF DISTILLATION USING INTERMEDIATE FEATURES BEFORE/AFTER APPLYING GUIDED ATTENTION.

Loss function	*Before*	*After*
$0.90 \cdot KL_{logit} + 0.10 \cdot CE_{sup} + 0.00 \cdot KL_{f_1} + 0.00 \cdot KL_{f_2}$	46.35	43.72
$0.30 \cdot KL_{logit} + 0.10 \cdot CE_{sup} + 0.30 \cdot KL_{f_1} + 0.30 \cdot KL_{f_2}$	**49.78**	**45.32**
$0.36 \cdot KL_{logit} + 0.10 \cdot CE_{sup} + 0.36 \cdot KL_{f_1} + 0.18 \cdot KL_{f_2}$	47.59	43.87
$0.18 \cdot KL_{logit} + 0.10 \cdot CE_{sup} + 0.36 \cdot KL_{f_1} + 0.36 \cdot KL_{f_2}$	49.77	43.55

In our investigation, we conduct a comprehensive evaluation of several loss functions, including Cross-Entropy (CE), Kullback-Leibler (KL) divergence, Mean Squared Error (MSE), and Mean Absolute Error (MAE), to ascertain the optimal configuration for our model. Furthermore, we investigated the coefficients of these components to determine their respective impacts on model performance. As a default, we use supervised loss as the CE and only change the remaining loss functions. Finally, with the config mentioned in Section IV.1, our best accuracy is 54.17%. Specifically, accuracy for "color" question is 53.12%, for "number" question is 42.79%, for "where" question is 44.96%, and for "what" question is 63.80%.

Table IV
THE COMPARISON BETWEEN THE MOVE AND THE STUDENT MODEL ON THE NUMBER QUESTION.

Class	Student (%)	MoVE (%)	Class Distribution (%)
One	23.33	**26.67**	6.76
Two	**53.73**	52.24	30.18
Three	**45.76**	44.07	26.58
Four	43.75	**51.04**	21.62
Five	30.30	30.30	7.43
Six	**25.00**	18.75	3.60
Seven	0.00	**40.00**	1.13
Eight	16.67	16.67	1.35
Nine	0.00	**50.00**	0.45
Ten	0.00	0.00	0.90
Total	42.79	**44.14**	100

b) Mixture of experts results

We implement a rule-based gate module and three classifiers, each tailored to a specific category of question types: (1) number, (2) color, and (3) a combined category for "what" and "where" questions. This architectural configuration is designed to optimize the model's performance across the diverse spectrum of question types encountered within the dataset.

In the ViVQA dataset, the rule-based gate classifier reaches 99.25%. Employing a rule-based module ensures both high accuracy and rapid execution speeds, thereby reducing the necessity for deploying deep learning models in the gate module.

Regarding accuracy improvements, the number classifier's accuracy increased by 1.35%, and the color classifier's accuracy increased by 0.48%. The entire system reached 54.37% on the whole dataset, while the total number of parameters increased by only 3.1% compared to the student model.

We conducted a comparative analysis between the student model and the MoVE system, focusing on two specific domains:

- **Number Questions**: As evidenced in Table IV, the MoVE system demonstrated capability in resolving certain questions labeled as "seven" and "nine", which were previously unsolvable by the student model. However, a notable decrement in MoVE's performance is observed within some classes, which constitute a significant portion of the dataset for number-related questions. This discrepancy has led to a marginal improvement in the overall accuracy of the system. Upon examining the distribution of classes, it is observed that the MoVE model achieves substantial advancements in performance for classes characterized by a lower frequency of samples.
- **Color Questions**: According to Table V, MoVE's performance was found to be inferior to that of the student model in merely three out of ten classes ("black", "purple", and "green"). The performance in the remaining classes exhibited a consistent improvement, barring the unchanged outcomes for "blue" and "white." Despite these gains, the improved classes do not represent a substantial fraction of the color-question dataset, resulting in only a slight enhancement in accuracy. Among the eight classes demonstrating varied performance levels, the MoVE model outperformed the student model in five classes, as opposed to the three classes where the student model excelled.

These findings provide a foundational basis for directing future efforts towards refining the MoVE system, with an aim to address the identified limitations and bolster the accuracy of the model across both the number and color question domains.

Table V
THE COMPARISON BETWEEN OUR MOVE AND THE STUDENT MODEL ON THE COLOR QUESTION.

Class	Student (%)	MoVE (%)	Class Distribution (%)
Orange	38.10	**40.48**	6.72
Black	**37.50**	32.14	8.96
Red	65.33	**68.00**	12.00
Brown	65.28	**66.67**	11.52
Purple	**58.54**	56.10	6.56
White	47.06	47.06	13.60
Yellow	59.42	**62.32**	11.04
Gray	31.43	**40.00**	5.60
Blue	52.70	52.70	11.84
Green	**57.89**	55.26	12.16
Total	53.12	**53.60**	100

3. Overall Results

Based on Table VI, training the student model with knowledge distillation improved accuracy by 7.07% compared to the student model without distillation. Furthermore, the student model with knowledge distillation reduced nearly 65 million parameters while only decreasing accuracy by 6.59% when compared to the teacher model.

These two findings demonstrate the successful application of knowledge distillation techniques to the VQA task. Moreover, implementing the mixture of experts approach helped improve the student model's predictive capabilities while only marginally increasing the number of parameters (approximately 2.9% increase compared to the student model). Further details on this improvement are provided in Section IV.2.b. However, since MoVE's improvements were primarily concentrated in classes with low distribution in the dataset, the overall results showed only modest gains.

Table VI
OVERALL RESULTS BETWEEN MOVE MODEL, STUDENT MODEL FROM SCRATCH, STUDENT MODEL WITH KNOWLEDGE DISTILLATION AND ORIGINAL MODEL.

Model	No. Parameters (M)	Accuracy (%)
Teacher model	~269	60.76
Student model (w/o knowledge distillation)	~204	47.10
Student model (w/ knowledge distillation)	~204	54.17
MoVE model	~210	54.37

Although the teacher model remains the most accurate, with an accuracy of 60.76%, its large size and resource demands make deployment challenging. Our distilled student model, and subsequently the MoVE system, aim to provide a practical trade-off between accuracy and efficiency by reducing model size. Moreover, the mixture-of-experts system further improves accuracy on certain challenging question categories with minimal additional parameters.

V. DISCUSSION

We propose a method called Distilling Step-by-Step with Multiple Rationales for extracting rationales from large language models (LLMs) to provide informative supervision in training small task-specific models. Our results demonstrate that this approach outperforms both traditional LLMs and other methods that use only rationales. Moreover, it reduces the need for extensive training datasets by replacing human-generated rationales. Distilling Step-by-Step proposes a resource-efficient training-to-deployment paradigm compared to existing methods. Further studies affirm the generalizability and the strategic design choices of our method. Below, we discuss the limitations, future directions, and ethical considerations of our work.

Our approach has several limitations. Firstly, it requires the use of the GPT-3.5 API, which is not only limited but also incurs costs. Secondly, although rationales significantly enhance the performance of our task-specific models, the evaluation of a rationale's quality whether it is effective or not remains unclear. Finally, despite the success of using LLM rationales, there is evidence that LLMs exhibit limited reasoning capabilities in more complex reasoning and planning tasks [28].

Looking ahead, there are several directions for future work with our method. First, we could explore the use of other LLMs, such as GPT-4.0 or Gemini, to potentially improve the quality of rationales, anticipating that newer models might offer enhanced capabilities. Second, we aim to investigate how the quality of rationales impacts the performance of task-specific models. Finally, while we have observed success in using LLM rationales for textual entailment tasks, we plan to extend this method to tasks like image description to enhance both the classification and description of images using similar strategies.

It is important to acknowledge that the behavior of our smaller downstream models may inherit biases from the larger teacher LLMs. We anticipate that advancements in reducing antisocial behaviors in LLMs could be adapted to enhance the ethical performance of smaller language models.

VI. CONCLUSION

In this study, we introduce a Mixture of ViVQA Experts system (MoVE) that takes in visual-textual representations from image-question pairs and routes them to specialized expert classifiers to resolve three primary question types that are frequently encountered in visual question answering datasets - namely $color$, $number$, and $other$ questions. The gating module, which performs dynamic routing between expert modules, helps our overall VQA system better adapt

to distinct data domains posed by different question types, thereby improving answer prediction accuracy compared to single module baselines.

Based on the findings of this work, we envision several promising future research directions to further advance the Mixture of ViVQA Experts system. One clear extension is to develop more fine-grained expert modules for additional question types beyond *color* and *number* that capture niche domains requiring specialized reasoning. Another useful improvement is to apply knowledge distillation to the textual branch feature extractors in a teacher-student training framework in order to reduce model size and improve inference speed while retaining accuracy gains. This approach has the potential to facilitate the porting of multi-expert systems to platforms with limited resources. Finally, expanding the ViVQA mixture-of-experts formulation proposed here to multi-lingual settings can allow leveraging non-English training data. By developing question-type prediction modules for other languages and aligning polyglot data across mixture components, such systems can answer visual questions posed in multiple global dialects. We are of the opinion that advancing research in these directions can lead to a broader and more practical viability for VQA systems.

REFERENCES

[1] A. C. A. M. de Faria, F. d. C. Bastos, J. V. N. A. da Silva, V. L. Fabris, V. d. S. Uchoa, D. G. d. A. Neto, and C. F. G. d. Santos, "Visual question answering: A survey on techniques and common trends in recent literature," *arXiv preprint arXiv:2305.11033*, 2023.

[2] B. Xin, N. Xu, Y. Zhai, T. Zhang, Z. Lu, J. Liu, W. Nie, X. Li, and A.-A. Liu, "A comprehensive survey on deep-learning-based visual captioning," *Multimedia Systems*, vol. 29, no. 6, pp. 3781–3804, Dec 2023.

[3] q. yue, R. Feng, F. Matsuzawa, T. Arai, and K. Iwata, "A survey of visual reasoning," 2023.

[4] N. S. Nguyen, V. S. Nguyen, and T. Le, "Advancing vietnamese visual question answering with transformer and convolutional integration," *Computers and Electrical Engineering*, vol. 119, p. 109474, 2024.

[5] C. Yang, S. Feng, D. Li, H. Shen, G. Wang, and B. Jiang, "Learning content and context with language bias for visual question answering," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[6] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei, "VLMo: Unified vision-language pre-training with mixture-of-modality-experts," in *Advances in Neural Information Processing Systems*, 2022.

[7] A. D. Nguyen, T. Le, and H. T. Nguyen, "Combining multi-vision embedding in contextual attention for vietnamese visual question answering," in *Image and Video Technology*. Cham: Springer International Publishing, 2023, pp. 172–185.

[8] K. V. Tran, K. Van Nguyen, and N. L. T. Nguyen, "Bartphobeit: Pre-trained sequence-to-sequence and image transformers models for vietnamese visual question answering," in *2023 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE, 2023, pp. 1–6.

[9] V. Joshi, P. Mitra, and S. Bose, "Multi-modal multi-head self-attention for medical vqa," *Multimedia Tools and Applications*, Oct 2023.

[10] A. A. Yusuf, C. Feng, X. Mao, R. Ally Duma, M. S. Abood, and A. H. A. Chukkol, "Graph neural networks for visual question answering: a systematic review," *Multimedia Tools and Applications*, Nov 2023.

[11] A. Berthelier, T. Chateau, S. Duffner, C. Garcia, and C. Blanc, "Deep model compression and architecture optimization for embedded systems: A survey," *Journal of Signal Processing Systems*, vol. 93, no. 8, pp. 863–878, Aug 2021.

[12] H. Raj Khan, D. Gupta, and A. Ekbal, "Towards developing a multilingual and code-mixed visual question answering system by knowledge distillation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1753–1767.

[13] J. Wang, S. Huang, H. Du, Y. Qin, H. Wang, and W. Zhang, "Mhkd-mvqa: Multimodal hierarchical knowledge distillation for medical visual question answering," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 567–574.

[14] K. Q. Tran, A. T. Nguyen, A. T.-H. Le, and K. V. Nguyen, "ViVQA: Vietnamese visual question answering," in *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*. Shanghai, China: Association for Computational Lingustics, 11 2021, pp. 683–691.

[15] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6281–6290.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[17] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: a compact task-agnostic BERT for resource-limited devices," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 2158–2170.

[18] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4163–4174.

[19] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[20] M. Nguyen, H. Abbass, and R. McKay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, pp. 155–163, 12 2006.

[21] R. Ebrahimpour, E. Kabir, H. Esteky, and M. R. Yousefi, "View-independent face recognition with mixture of experts," *Neurocomputing*, vol. 71, no. 4, pp. 1103–1107, 2008, neural Networks: Algorithms and Applications 50 Years of Artificial Intelligence: a Neuronal Approach.

[22] V. Pahuja, J. Fu, and C. J. Pal, "Learning sparse mixture of experts for visual question answering," *arXiv preprint arXiv:1909.09192*, 2019.

[23] G. Liu, J. He, P. Li, S. Zhong, H. Li, and G. He, "Unified transformer with cross-modal mixture experts for remote-sensing visual question answering," *Remote Sensing*, vol. 15, no. 19, 2023.

[24] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers:

scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 1, jan 2022.

[25] J. Feng, E. Tang, M. Zeng, Z. Gu, P. Kou, and W. Zheng, "Improving visual question answering for remote sensing via alternate-guided attention and combined loss," *International Journal of Applied Earth Observation and Geoinformation*, vol. 122, p. 103427, 2023.

[26] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[27] L. Hou, C.-P. Yu, and D. Samaras, "Squared earth mover's distance-based loss for training deep neural networks," *arXiv preprint arXiv:1611.05916*, 2016.

[28] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, "Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change)," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Nguyen Tien Huy received his B.S. degree in Software Technology in 2010 and his M.S. degree in Computer Science in 2015, both from the University of Science, Ho Chi Minh City, Vietnam (VNU-HCM). He earned his Ph.D. in Information Science from the Japan Advanced Institute of Science and Technology (JAIST) in 2019. He is currently a lecturer in the Department of Computer Science, Faculty of Information Technology, VNU-HCM. His research interests include Natural Language Processing and Deep Learning, with a particular focus on intelligent systems and multimodal data analysis.

Email: ntienhuy@fit.hcmus.edu.vn

Tung Le is a Lecturer of Information Technology at the University of Science, Ho Chi Minh City, Vietnam (HCMUS). He earned his B.S. degree in Computer Science from the Honour Program at the University of Science, Vietnam National University, Ho Chi Minh City, in 2012, and his M.S. degree in Information Science from the Japan Advanced Institute of Science and Technology (JAIST) in 2018. He completed his Ph.D. in Information Science at JAIST in 2021. His research interests include information retrieval, natural language processing, visual question answering, and multi-modal systems.

Email: tungle@fit.hcmus.edu.vn

Huynh Hoang Huy received the B.S. degree in Computer Science from the University of Science, Ho Chi Minh City, Vietnam (VNU-HCM), in 2021. He has completed a Master's degree at the University of Science. His research focuses on deep learning and its advancement toward real-world implementation.

Email: hhoanghuy@fit.hcmus.edu.vn

Biomedical Entity-Aware Semantic Role Labelling via Model Merging

Hoai-Duc Tuan-Nguyen[1,2]

[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Correspondence: Hoai-Duc Tuan-Nguyen. Email: tnhduc@fit.hcmus.edu.vn
Communication: received 09/05/2025, revised 03/06/2025, accepted 18/06/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1368

***Abstract*: Semantic Role Labeling (SRL) is essential for extracting structured knowledge from biomedical texts, yet it remains challenging due to the complexity of biomedical terminology. This paper introduces a novel method that enhances biomedical SRL by directly integrating Named Entity Recognition (NER) knowledge using model merging. NER plays a crucial role in anchoring domain-specific arguments, allowing the model to better distinguish semantic roles in biomedical contexts. Unlike prior approaches that rely on multitask learning or syntax-aware features, our method merges pretrained SRL and NER models in a way that preserves entity knowledge while aligning with semantic role structures. Experimental results demonstrate that our approach surpasses state-of-the-art multitask and syntax-aware SRL models in both accuracy and training speed. The integration of biomedical entity knowledge proves especially effective for predicates with high proportions of entity-based arguments, highlighting the value of direct NER-to-SRL knowledge transfer in advancing biomedical semantic understanding.**

Keywords: *Model merging, NER, NLP, SRL.*

I. Introduction

Biomedicine is a field that applies biotechnology to human healthcare. It plays an important role in diagnosing and treating diseases. In recent years, research in biomedicine has grown rapidly, leading to a vast amount of knowledge being stored in scientific texts. However, the increasing volume of biomedical literature makes it impossible for humans to manually process all this information [1]. To make full use of this knowledge, automated methods are needed to extract meaningful information efficiently [2, 3].

A key challenge in biomedical text mining is extracting structured knowledge, which requires identifying the relationships between actions (predicates) and the entities involved in those actions (arguments). A method to achieve this is through Semantic Role Labeling (SRL), a technique that helps computers recognize who is doing what to whom in a sentence. SRL is based on Predicate Argument Structure (PAS), where the predicate represents the main verb describing an event or action, and the arguments provide the key details related to that action [4]. For example, in the sentence "Aspirin inhibits platelet aggregation", the predicate is inhibits, while Aspirin is the agent (the entity performing the action), and platelet aggregation is the affected entity (the patient of the action).

Although SRL has been widely studied in general texts, applying it to biomedical literature presents new challenges. Biomedical sentences tend to be more complex, with specialized terminology and intricate sentence structures that make it harder for computers to recognize arguments correctly [5]. Another difficulty is the lack of annotated training data, which limits the ability of machine learning models to learn from examples [6]. Because of these challenges, adapting SRL models for biomedical texts requires special techniques, such as fine-tuning pre-trained language models with biomedical-specific knowledge [7].

One important factor in improving SRL for biomedical texts is the use of biomedical entities, such as proteins, DNA, RNA, cell lines, and cell types 5. These entities often appear as arguments in PAS, and recognizing them correctly helps improve the accuracy of SRL [8]. Therefore, we propose an improved approach that combines biomedical entity knowledge with a pre-trained language model to enhance SRL performance on biomedical text. As SRL focuses on identifying arguments within PAS, recognizing these domain-specific entities provides additional context that aids in distinguishing arguments and assigning their correct semantic roles.

To evaluate our approach, we conducted experiments using the PASBio+ corpus, a comprehensive biomedical SRL dataset [6]. Experimental results demonstrate that our method of incorporating biomedical entity knowledge enhances SRL performance, outperforming both multitask

learning and syntax-aware SRL approaches tested on the same dataset, as well as the baseline model that lacks entity information. These findings demonstrate the effectiveness of integrating entity knowledge in SRL for biomedical text mining and highlight a promising direction for future research in the field. While our method is evaluated in a biomedical context, the model merging strategy is domain-agnostic and can be extended to other domains in future work. We explicitly scope this study to biomedical predicates, where entity-based argument structures are prominent and well-defined.

This paper is structured as follows: Section 1 introduces the significance of the SRL task and provides an overview of our proposed approach. Section 2 provides background on PAS, contrasts general and biomedical domains, and reviews existing datasets. Section 3 discusses related work, emphasizing the role of entity knowledge in SRL. Section 4 details our proposed model, while Section 5 presents experimental results and analysis. Finally, Section 6 summarizes key contributions and suggests directions for future research.

II. Background

In NLP, Predicate-Argument Structure (PAS) represents how a predicate (the main verb of a sentence that describes an action or state) connects to its arguments (entities involved in that action or state). Each argument is assigned a semantic role, which explains its function in relation to the predicate.

For example, consider the sentence: "The TP53 gene encodes a tumor suppressor protein in human cells". Here, the PAS consists of the predicate "encode" and two arguments:

- Arg0: "the TP53 gene" (semantic role: gene or RNA)
- Arg1: "a tumor suppressor protein in human cells" (semantic role: gene product)

PAS is essential for extracting meaningful biomedical relationships, such as identifying which drugs interact with specific diseases or which proteins regulate particular biological processes. Therefore, understanding PAS helps machines capture the main meaning of a sentence. In general-domain NLP, several PAS-annotated datasets exist, such as FrameNet [9], VerbNet [10], and PropBank [11], with PropBank providing the most detailed argument structures.

In biomedical texts, PAS works differently from general-domain PAS, mainly because the roles of arguments change. As exemplified in Table I, the verb mutate has different meanings depending on the context. In general texts, it describes a change, focusing on who or what caused the change and the states before and after [12]. However, in biomedical texts, it describes genetic mutations, where arguments refer to the mutation site, the affected gene, and the resulting molecular and phenotypic changes [5].

Table I
BIOMEDICAL AND GENERAL ARGUMENT FRAMES OF THE PREDICATE MUTATE.

Biomedical PAS (PASBio)	**General PAS (PropBank)**
Mutate = Exhibit a phenotype • Arg0: Physical location where mutation happen (exon, intron) • Arg1: Mutated entity (gene) • Arg2: Changes at molecular level • Arg3: Changes at phenotype level	Mutate = Changing state, morphing • Arg0: Causer of change • Arg1: Entity undergoing mutation • Arg2: End state • Arg3: Start state

Because of such differences, specialized biomedical PAS datasets have been developed. Unlike general-domain PAS resources, which define argument structures for all verbs, biomedical PAS datasets focus only on key biomedical verbs, such as mutate, encode, express, catalyse,... However, researchers have not yet agreed on a standard list of biomedical predicates. Some of the most well-known biomedical PAS datasets are:

- BioProp: Comprising 1,635 sentences extracted from 500 biomedical papers, this dataset adopts general-domain PAS structures from PropBank [1]. Therefore, its argument roles are not fully adapted to biomedical texts.
- GREC: Containing 1,489 sentences from 677 biomedical abstracts, GREC improves upon BioProp by defining biomedical-specific argument roles [3]. However, it does not explicitly link predicates and arguments, instead treating them as independent elements.
- BioVerbNet : With 521 annotated sentences, BioVerbNet extends VerbNet by incorporating missing biomedical verbs [13]. However, it suffers from limited annotations and retains general-domain PAS structures, making it less suitable for biomedical SRL.
- PASBio+: PASBio+ is the most comprehensive dataset [6]. Unlike BioProp and BioVerbNet, PASBio+ defines argument frames specifically tailored for biomedical predicates [5]. Unlike GREC, PASBio+ explicitly links each argument to its predicate. PASBio+ is also larger than both BioProp and GREC, making it more effective for training biomedical SRL models.

In Semantic Role Labeling (SRL), selecting the right argument frameset is crucial for accurate sentence understanding. We choose PASBio+ for its biomedical-specific argument roles, explicit predicate-argument relationships, and larger dataset size, making it more effective for SRL training than other biomedical PAS resources.

III. RELATED WORKS

Initial efforts in biomedical Semantic Role Labeling (SRL) focused on adapting general-domain SRL techniques to biomedical corpora. BIOSMILE was among the earliest systems tailored for biomedical text, employing PropBank-style annotation [12] and training a maximum entropy model to classify semantic roles of biomedical predicates [14]. While BIOSMILE achieved reasonable performance, it remained constrained by a general-domain argument frameset (BioProp), limiting its applicability in biomedical contexts. A later adaptation expanded the predicate set from 30 to 82 verbs and incorporated syntactic parsing-based features, improving role classification but still struggling to generalize to unseen biomedical text due to its continued reliance on general-domain argument frames [15]. To overcome this limitation, BelSmile extended BIOSMILE by replacing the maximum entropy model with Conditional Random Fields (CRF) and incorporating domain-specific linguistic rules to refine SRL outputs, particularly for biological expression language extraction [16]. This approach introduced biomedical-specific constraints, reducing reliance on general-domain framesets; however, it remained rule-based, requiring extensive domain expertise and lacking scalability across diverse biomedical subdomains.

To improve efficiency in biomedical SRL, a resource-efficient SRL approach was introduced, leveraging Markov Logic Networks (MLN) for collective learning and integrating tree-pruning and argument candidate identification to reduce memory usage and training time [17]. This method enhanced accuracy by preventing duplicate and overlapping arguments; however, it struggled with less common argument structures, particularly those involving discourse markers or mistakenly pruned arguments.

Beyond these efforts, several studies have highlighted the critical role of Named Entity Recognition (NER) in SRL, particularly in refining argument classification. While NER alone does not capture full predicate-argument structures, it serves as a foundation for anchoring key arguments and refining semantic interpretations [18]. A systematic review emphasized that NER aids SRL by enhancing entity identification, which in turn improves predicate-argument role classification [19]. Biomedical SRL resources such as GREC [3], PASBio [5], and PASBio+ [6] integrate Named Entity (NE) categorization within SRL annotations, explicitly labeling entities such as DNA, proteins, and biological processes within semantic arguments. This explicit entity annotation ensures that machine learning models can leverage domain-specific entity types to improve SRL. Studies also demonstrate that named entity categorization aids in determining the types of phrases that fill specific semantic roles, reinforcing its necessity in accurate and domain-adapted SRL applications [3]. A recent study further revealed that 63

Advancements in deep learning-based SRL have increasingly sought to integrate NER knowledge. A study by [20] explored the integration of Entity Recognition (ER) as an auxiliary task to enhance Semantic Role Labeling (SRL) performance in low-resource, conversational domains. By combining multi-task learning with active learning, the proposed model leveraged shared semantic information between ER and SRL, where ER provides entity type cues that help regularize SRL predictions. Their experiments on Indonesian dialogue data demonstrated that multi-task learning not only improved SRL accuracy but also reduced the need for annotated data, showing the utility of ER in refining argument role classification through shared representation learning. A bidirectional LSTM-CRF model was designed to jointly perform NER and SRL in medical documents, leveraging biomedical NER embeddings to capture contextual relationships between predicates and arguments [8]. However, due to the absence of a large language model (LLM) architecture, the model struggled with ambiguities in role classification, as biomedical entities often exhibit complex interdependencies that classic deep learning architectures fail to capture.

While many recent models have explicitly incorporated syntax-based SRL approaches, they implicitly used NER knowledge. A biomedical SRL model trained on PASBio+ leveraged entity-centered SRL annotations to improve argument classification [21]. Additionally, the model integrated BioBERT as a contextual encoder, leveraging entity-aware embeddings to enhance semantic role representations in biomedical text. Further research has demonstrated that constituency parsing can serve as a structural foundation for SRL, with named entities often appearing as constituents within parse trees, thereby aiding predicate-argument extraction [22]. Moreover, its TreeCRF-based structured model followed NER span-detection methods, reinforcing the structural similarities between predicate-argument spans in SRL and entity spans in NER. These findings suggest that advancements in NER span-based modeling could benefit SRL, particularly in argument role assignment.

Given their close interdependence, [23] introduced a unified framework for low-resource sequence labeling that jointly learns NER and SRL, using a shared augmented language representation. By incorporating named entity information within the input format and adapting model parameters based on underperforming instances, the approach significantly improved SRL performance. These results highlight the value of entity cues in guiding role assignment, demonstrating that improvements in NER can

contribute meaningfully to more accurate SRL, particularly in low-resource settings.

Despite widespread recognition of NER's benefits in SRL, current methodologies have limitations (Table II). Specifically, they primarily incorporate entity knowledge indirectly, either via entity-centered SRL corpora or through structural similarities between NER and SRL spans. While some neural models integrate NER and SRL, they remain limited to conventional neural network architectures, lacking the representational power of modern LLMs. Recent work in 2025 has increasingly embraced model merging as a scalable, modular alternative to full fine-tuning, especially in sensitive domains like healthcare. For example, the PatientDx framework shows that merging pre-trained models can preserve patient privacy while improving performance on clinical prediction tasks [24]. Likewise, a domain-adaptive strategy demonstrates how merging general and biomedical LLMs enhances downstream performance without retraining [25]. The Modular Clinical SLM framework illustrates how pre-instruction tuning and task-specific alignment through merging can streamline adaptation for clinical use cases [26]. Complementing these, STAR method introduces spectral truncation and rescaling to mitigate instability during merging and optimize transferability [27]. Motivated by these advances, our work proposes a novel approach that leverages model merging to directly integrate NER with SRL in an LLM-based architecture. This enables more precise argument extraction in biomedical texts by leveraging entity-aware contextual representations, positioning our method at the forefront of emerging model integration techniques.

IV. Method

1. Foundation model selection

Our approach involves merging a baseline SRL model with a pre-existing NER model via confidence-based model merging. This merging strategy operates at the level of task vectors, which represent layer-wise weight differences from a shared base model. As a result, both constituent models (SRL and NER) must be derived from the same underlying architecture and initialization to ensure compatibility during the merging process. The publicly available NER model we leverage, achieving an F1 score of up to 93.04

While alternative biomedical language models such as PubMedBERT [29] and ClinicalBERT [30] have been introduced, they are not suitable for our merging mechanism in this context. While PubMedBERT has demonstrated strong performance across various biomedical NLP tasks, its direct application to SRL tasks within the biomedical domain hasn't been featured in the literature. ClinicalBERT, on the other hand, is fine-tuned on MIMIC-III clinical notes and is therefore optimized for processing clinical narratives, rather than scientific biomedical literature.

In contrast, BioBERT has been adopted and validated on biomedical SRL [21]. Moreover, the availability of high-quality, fine-tuned BioBERT-based NER models made it the suitable choice for our model merging setup without retraining an auxiliary model from scratch. We therefore selected BioBERT as the foundation model for our SRL approach.

2. Constituent models for merging

We use the NER model provided by BioBERT [7]. As for the SRL model, we fine-tune the BioBERT pre-trained model to identify semantic roles in biomedical text by leveraging contextual embeddings. A token-level classification layer with a softmax activation function is added to predict predicates and argument roles based on the model's final hidden states. The model is optimized using cross-entropy loss, which aligns predicted probabilities with true labels.

3. Model merging method

Given these two constituent models, the goal is to create a merged SRL model that leverages beneficial knowledge from the NER model to enhance SRL performance. Researchers typically develop models only for their main task and reuse pre-trained auxiliary models from other studies, which are usually shared without their original training data. Therefore, our study focuses on a merging solution that does not require training data for the auxiliary task (in this case, the NER task). The process must avoid retraining on the original NER datasets and instead rely on unlabeled samples for optimization.

The merging process is based on the concept of task vectors [31], which represent task-specific adaptations at both the model and layer levels. Each task vector is obtained by subtracting the weights of a pre-trained foundation model from those of a fine-tuned model. Formally, let $\theta_{BioBERT}$ denote the weights of the foundation model before fine-tuning (i.e., BioBERT), and let θ_{SRL} and θ_{NER} represent the weights of the fine-tuned SRL and NER models, respectively. The overall task vectors representing SRL and NER are specified in Equations (1):

$$\begin{aligned} T_{SRL} &= \theta_{SRL} - \theta_{BioBERT} \\ T_{NER} &= \theta_{NER} - \theta_{BioBERT} \end{aligned} \tag{1}$$

At the layer level, the task vector for the l-th layer, denoted as $T_{task}^{(l)}$, is computed similarly by subtracting the

Table II
BIOMEDICAL AND GENERAL ARGUMENT FRAMES OF THE PREDICATE MUTATE.

Theme	Works	Key Contributions	Limitations
General SRL Adapted to Biomedical	BIOSMILE [14]	Adapted PropBank-style SRL to biomedical verbs using MaxEnt models.	Relied on general-domain frames; limited generalization to unseen biomedical text.
	Extended BIOSMILE [15]	Expanded verb coverage, added syntactic parsing features.	Still used general-domain PAS frames; poor performance on novel biomedical inputs.
Biomedical SRL	BelSmile [16]	Introduced domain-specific linguistic rules; used CRFs.	Rule-based; lacked scalability; required domain expertise.
	MLN-based SRL [17]	Efficient inference using collective reasoning; avoided duplicate and overlapping arguments.	Struggled with less frequent structures (e.g., discourse markers).
NE in SRL corpora	GREC, PASBio, PASBio+ [3, 5, 6]	Provided SRL corpora with explicit entity annotations (e.g., DNA, protein).	May suffer from label granularity or corpus size constraints.
	GENEVA [28]	Showed 63% of argument roles are entity-based, confirming NER's importance.	Entity overlap still poses challenge.
Joint or Integrated NER-SRL Models	Multi-task ER-SRL [20]	Used ER as auxiliary task to regularize SRL in low-resource conversational data.	Low-resource setting; not LLM-based.
	BiLSTM-CRF (Joint) [8]	Joint model for NER and SRL using medical NER embeddings.	Lacked LLM backbone; struggled with complex biomedical interdependencies.
	TreeCRF model [22], Syntax-aware SRL [21]	Incorporated syntactic and span-based NER insights into SRL.	Indirect NER use; relies on parse tree quality.
	Unified ER+SRL framework [23]	Joint low-resource learning using shared representations and input augmentation.	Not optimized for high-resource biomedical texts.

weight matrix of the foundation model at layer l, $\theta^{(l)}_{BioBERT}$, from that of the fine-tuned model ($\theta^{(l)}_{SRL}$ or $\theta^{(l)}_{NER}$).

$$\begin{aligned} T^{(l)}_{SRL} &= \theta^{(l)}_{SRL} - \theta^{(l)}_{BioBERT} \\ T^{(l)}_{NER} &= \theta^{(l)}_{NER} - \theta^{(l)}_{BioBERT} \end{aligned} \tag{2}$$

The layer-wise task vectors in Equations (2) capture the localized layer-specific modifications introduced during fine-tuning, enabling a more granular integration of task-specific knowledge when merging models. The merging of the SRL model with the NER model is achieved by combining their respective task vectors $T^{(l)}_{SRL}$ and $T^{(l)}_{NER}$ at each layer using a weighted sum. This process forms a unified task vector $T^{(l)}_{Merged}$ for the l-th layer of the merged model, which captures both semantic role labeling and named entity recognition knowledge:

$$T^{(l)}_{Merged} = T^{(l)}_{SRL} + \lambda^{(l)} T^{(l)}_{NER} \tag{3}$$

In Equation (3), $\lambda^{(l)}$ is the layer-specific coefficient for the NER task vector at the l-th layer, determining its contribution to each layer of the merged SRL model ($0 \leq \lambda^{(l)} \leq 1$). The SRL task vector's coefficient is 1, as it is the primary task in our study. Lower layers, capturing general features, tend to have lower coefficients, while higher layers, capturing task-specific features, receive higher coefficients [32].

The merged model's weights for each layer are computed by Equation (4):

$$\theta^{(l)}_{Merged} = \theta^{(l)}_{BioBERT} + T^{(l)}_{Merged} \tag{4}$$

Figure 1 illustrates an example of this merging process for layers 1 and 2. This layer-wise approach ensures that each layer incorporates the right amount of NER information while preserving the overall structure of the SRL model. A critical challenge is determining the optimal layer-wise coefficients $\lambda^{(l)}$ to effectively leverage NER knowledge for improving SRL performance. This is particularly difficult since the auxiliary model (in our case, the NER model) is typically available through other studies, but its original training data is often inaccessible. To overcome this, we propose Maximum-Confidence optimization.

4. Maximum-Confidence optimization

Entropy has been shown to correlate strongly with prediction loss. Grouping test samples by entropy levels reveals a consistent trend where higher entropy aligns with increased loss, further supported by a Spearman correlation of 0.87 between entropy and loss, confirming entropy as a reliable proxy for loss minimization [32]. These findings suggest that entropy minimization can effectively guide optimization in multi-task learning when access to original training data is unavailable.

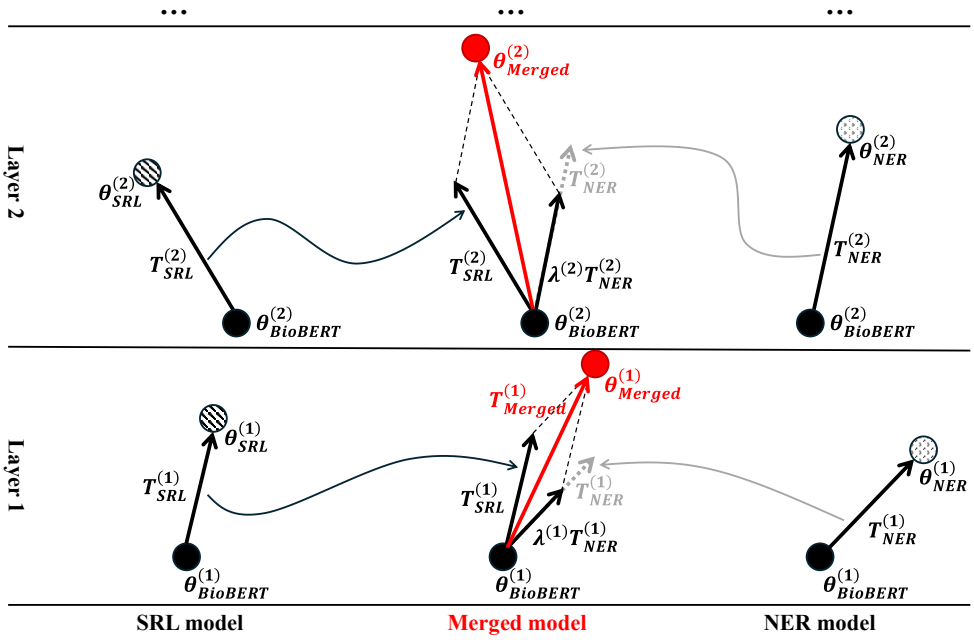


Figure 1. An example of layer-wise merging, showing how the NER task vector contributes to layers 1 and 2 of the merged SRL model.

However, our study finds that prediction confidence exhibits an even stronger correlation with loss. Specifically, when evaluating the NER model on the JNLPBA corpus [33], we observe a Spearman correlation of -0.91 between confidence and loss, surpassing the 0.87 correlation between entropy and loss reported by [32]. These results suggest that higher confidence consistently corresponds to lower prediction loss, reinforcing confidence as a more reliable indicator of model performance.

Given this stronger correlation, confidence maximization presents a more effective optimization objective than entropy minimization. Maximizing confidence ensures that the model produces more certain predictions, which typically correlate with higher accuracy. By directly encouraging high-confidence predictions, optimization can better reduce loss, leading to improved multi-task learning outcomes. Formally, the optimization function is defined as:

$$\underset{\{\lambda^{(l)}\}_{l=1}^{L}}{\operatorname{argmin}} \left(\sum_{x_i \in D_{SRL}} \mathcal{L}_{SRL}\left(x_i; \{\lambda^{(l)}\}_{l=1}^{L}\right) + \beta \sum_{x_k \in D_{NER}} (1 - C_{NER}(x_k)) \right) \quad (5)$$

In Equation (5):

- L is the total number of model layers.
- D_{SRL} is the training data for the main task (SRL).
- D_{NER} is the unlabeled test dataset for the auxiliary task (NER).
- $C_{NER}(x_k)$ represents the NER prediction confidence for the test sample x_k.

The layer-wise coefficients $\lambda^{(l)}$ are optimized for the main task based on loss, as SRL-annotated training data is available, while Maximum-Confidence optimization is used for the auxiliary task to compensate for the absence of original NER training data. The hyperparameter β, empirically set to 0.5, regulates the relative importance of the NER task in the optimization.

Figure 2 summarizes our overall framework. The workflow begins by computing task vectors that capture how the NER and SRL models differ from the shared BioBERT foundation. Each model's adaptations are extracted layer by layer, forming task-specific representations. These vectors are then passed to the *Layer-wise merging* component, where they are combined using adjustable coefficients that control the influence of NER on the SRL model. Next, in the *Merged weight computation* component, the merged task vectors are added back to the BioBERT base to generate the parameters of a new, integrated model. This merged model is evaluated using the SRL loss function, which compares its predictions against gold-standard SRL annotations. Simultaneously, the model's NER output confidence is measured. Finally, the *Coefficient tuning* component adjusts the merging coefficients based on both the SRL loss and NER confidence. These updated coefficients

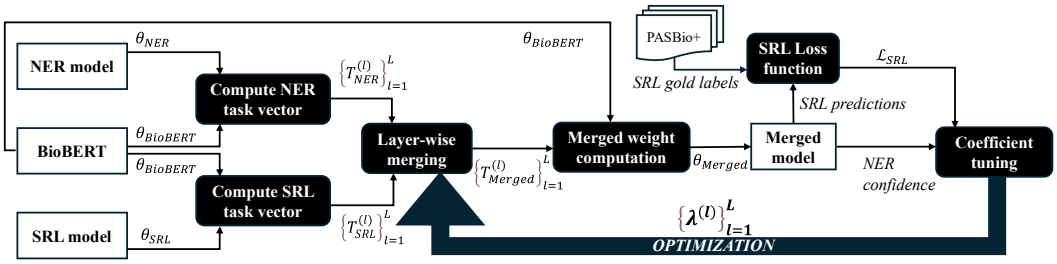


Figure 2. Overview of the confidence-based model merging framework.

are fed back into the merging process, completing the loop and iteratively refining the merged model's performance.

V. Experiments

1. Experimental data and setting

We assess NER models using the JNLPBA corpus [33]. For SRL evaluation, multiple datasets were reviewed, though most were found lacking. We ultimately chose PASBio+ [6], a specialized biomedical corpus featuring 35 core predicates. PASBio+ was selected based on several strengths. It is built specifically for the biomedical domain, ensuring task relevance. It offers clearly annotated argument structures grounded in domain-specific verb semantics. It provides gold-standard predicate–argument structures, enabling rigorous evaluation of SRL outputs. Additionally, it has been recently expanded to include named entity annotations, supporting both SRL and NER assessments.

- Biomedical Focus: Drawn exclusively from domain-specific literature, PASBio+ aligns closely with our study's subject matter.
- Adequate Scale: With more than 15,000 annotated sentences, the dataset provides sufficient volume for training. In contrast, smaller sets like BioVerbNet [13] and GREC [3], each under 1,500 sentences, risk data sparsity and performance degradation.
- Specialized Argument Structures: The argument structures, designed by domain experts, reflect biomedical semantics in detail. This is a key advantage over resources such as [1] or BioVerbNet [13], which adapt general linguistic frames to biomedical texts.

SRL evaluation: We evaluate the performance of our entity-aware SRL model, which is optimized via confidence-based model merging (Section IV), on biomedical text. The model is compared against the following models, all evaluated on the same dataset:

- A baseline SRL model without NER integration.
- An entity-aware SRL model using traditional multitask learning.
- Another entity-aware SRL model that uses entropy-based Adamerging to integrate NER knowledge [32].
- SRL models that incorporate syntactic knowledge instead of NER, achieving state-of-the-art results on PASBio+ [21].

This setup enables a comparison not only between different methods of integrating NER knowledge but also between NER-based and syntax-based enhancements to SRL performance on the same biomedical dataset.

2. Experimental result and analysis

We evaluate our entity-aware SRL model, which integrates NER knowledge via confidence-based model merging. We compare our model against four baselines introduced in Section III. Experimental results show that our method outperforms the baselines in both SRL performance and training speed.

Table V presents the SRL and NER F1 scores across models. Our entity-aware SRL model achieves the highest SRL F1 and accuracy, indicating superior overall performance and predictive correctness across all sentences. In addition, it attains strong precision (93.86%) and recall (94.31%), reflecting both accurate role identification and comprehensive argument span coverage.

Although the NER F1 score slightly drops compared to multitask learning, the performance remains adequate. This suggests that while the merged model does not preserve the full standalone NER performance, it retains sufficient entity-related information to improve SRL accuracy, as evidenced by: (i) the model achieving the improved SRL F1 score, and (ii) maintaining a moderate NER F1 score (76.00%), indicating that a substantial amount of NER knowledge remains usable after merging.

The models were tested on a server with an Intel(R) Xeon(R) CPU @ 2.00 GHz, a Tesla T4 GPU with 15 GB VRAM, and 12.7 GB of system RAM. Table III compares training time. Our method is also more efficient, with training time nearly halved compared to multitask learning and lower than entropy-based merging. This improvement

is attributed to the computational simplicity of confidence score calculation over entropy estimation. Additionally, in terms of average processing time per sentence, our model achieves the lowest latency at 0.685 seconds per sentence, compared to 0.768 for multitask learning and 0.826 for entropy-based merging. This demonstrates that our confidence-based merging method is not only efficient in total training time but also scalable for real-time or large-scale applications where per-sentence inference speed matters.

Table III
TRAINING TIME AND INFERENCE SPEED COMPARISON (IN SECONDS).

Metric	Multitask SRL+NER	Entropy Merging	Confidence Merging
Training time	7136	3539	2940
AVG time per sentence	0.768	0.826	0.685

We examined the effect of NER knowledge across individual predicates. For a small subset of predicates, performance declined after merging, as shown in Table IV.

Table IV
PREDICATES WITH F1 DECLINE POST-MERGING.

Predicate	F1 after	F1 before	after - before
result	0.8608	0.9898	-0.1290
lead	0.4822	0.7727	-0.2905
abolish	0.7702	0.7806	-0.0104
begin_2	0.9657	0.9787	-0.0130
translate_3	0.9806	0.9832	-0.0026
begin_1	0.9893	0.9904	-0.0011

Importantly, according to the PASBio predicate–argument frameset [5], each predicate is associated with two to five arguments, including both entity-based and non-entity-based arguments. Figure 3 illustrates an example of this structure. All the predicates in Table IV lack entity-based arguments according to the GENIA taxonomy. Figure 4 illustrates this using the predicate *result*, whose arguments are not categorized as biomedical entities, leading NER information to contribute noise rather than guidance.

To better understand how entity knowledge contributes to SRL improvement, we grouped predicates (excluding those in Table IV) into four tiers based on their baseline SRL F1 scores (i.e., the F1 score of the SRL model before merging with the NER model): (a) below 50%, (b) 50% to below 80%, (c) 80% to below 90%, and (d) 90% and above (Fig 5). This partitioning reflects the assumption that lower-performing predicates generally have more room for improvement and are thus more likely to benefit from the integration of NER knowledge. We then calculated the proportion of entity-based arguments. For example, the predicate *alter* in Figure 3 has five arguments, three of which are entity-based, resulting in a proportion of $3/5 = 0.6$.

Figure 5 visualize the relationship between F1 gain and the proportion of entity-based arguments across these four tiers. A consistent pattern emerges: predicates with higher entity-based argument proportions, indicated by taller yellow columns, yield greater SRL performance improvement, as reflected by a larger gap between the blue lines (SRL F1 after merging with the NER model) and red lines (SRL F1 before merging with the NER model).

Experimental results also show that predicates with high ambiguity in entity class assignments tend to exhibit smaller F1 improvements, even compared to predicates with a lower proportion of entity-based arguments. This is likely because such ambiguity weakens the ability of entity types to signal semantic roles or distinguish between different arguments.

One type of ambiguity occurs when a single entity-based argument is annotated with multiple entity classes. For example, predicate *delete*'s `arg1` (the entity being removed) spans five different entity classes, resulting in lower F1 gains than predicates *disrupt*, *eliminate*, and *decrease_2*, which have the same proportion of entity-based arguments (Fig 5a). Similarly, predicate *skip*'s `arg2` is linked to both DNA and RNA classes and improves less than predicates *translate_2* and *decrease_1* (Fig 5d).

Another form of ambiguity arises when multiple arguments of the same predicate share the same entity class, making it difficult to leverage NER knowledge to distinguish between them. In predicate *recognize*, both `arg0` and `arg1` are associated with the protein class, leading to limited improvement (Fig 5a). In predicate *encode*, `arg0` (gene) and `arg1` (protein) fall under the same GENIA class and are often referred to using the same term in biomedical texts, reducing the benefit of NER (Fig 5b). Similarly, predicates *splice_2* and *truncate* involve multiple arguments tied to the RNA class, which hampers disambiguation and yields modest F1 gains (Fig 5c and d).

A distinct case involves entity-based arguments that only partially include an entity mention. For instance, predicate *block*'s `arg1` may describe a biological process that references a gene or protein, but the argument as a whole is not an entity. This limits the contribution of NER to SRL in such instances (Fig 5d).

In some cases, F1 gains are modest despite a high percentage of entity-based arguments due to ceiling effects, where performance is already near optimal. Predicate *tran-*

Table V
SRL AND NER PERFORMANCE METRICS ACROSS MODELS.

Metric	Baseline SRL	Multitask SRL+NER	Syntax-aware SRL	Entropy-based SRL	Confidence-based SRL
SRL Precision	82.70	92.41	89.52	44.38	93.86
SRL Recall	86.80	93.01	91.24	66.62	94.31
SRL F1	83.80	92.71	90.37	53.27	94.08
SRL Accuracy	73.46	86.41	82.43	66.62	88.83
NER F1	–	80.91	–	76.03	76.00

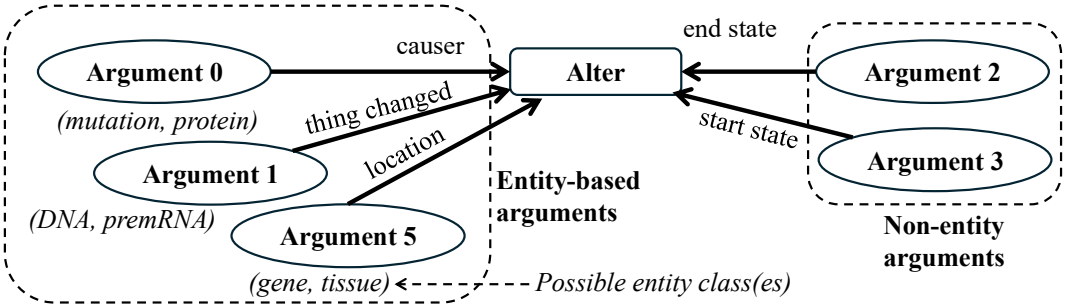


Figure 3. Example of PASBio predicate-argument structure with both entity-based and non-entity-based arguments.

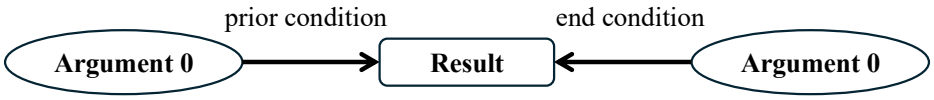


Figure 4. An example predicate (result) with no entity-based arguments.

scribe (Fig 5b) illustrates this: all arguments are entity-based, yet the baseline F1 is already very high.

Conversely, predicates like *proliferate* achieve greater F1 gains than predicates with a higher proportion of entity-based arguments. Only one of *proliferate*'s three arguments is entity-based. However, it appears as a predicate in 173 sentences, 49 of which contain only its `arg0` (i.e., 100% of the mentioned arguments in those cases are entity-based), and another 71 contain only `arg0` and `arg1` (i.e., 50% of the mentioned arguments are entity-based). This elevated frequency increases the practical influence of the entity-based argument beyond what the raw $1/3$ proportion suggests (Fig 5c).

Taken together, these findings suggest that entity-aware model merging is particularly effective when predicates are structurally aligned with biomedical entity distinctions, and that confidence-based merging provides a computationally efficient and semantically sound mechanism for integration.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper introduced a novel approach to enhancing Semantic Role Labeling (SRL) for biomedical texts by directly merging SRL and NER models using confidence-based merging optimization. The approach is specifically tailored for biomedical SRL, where domain-specific entities are prevalent and tend to align well with predicate-argument structures, making the integration of NER knowledge particularly effective . Our method significantly improves SRL performance on the PASBio+ dataset, outperforming state-of-the-art multitask and syntax-aware methods while requiring less training time. Empirical results confirm that biomedical entity knowledge plays a critical role in improving SRL, especially for predicates with high proportions of entity-based arguments. However, F1 gains vary depending on argument ambiguity, entity class overlap, and the completeness of entity span coverage.

We acknowledge that when predicates exhibit argument overlap in entity classes or span multiple semantic types, entity-based guidance may be less effective. Addressing such ambiguity is an important direction for future enhancement. For example, future works could explore fine-grained entity typing to replace coarse biomedical classes with more expressive representations that capture nuanced functional roles. Exploring adaptive entity representations

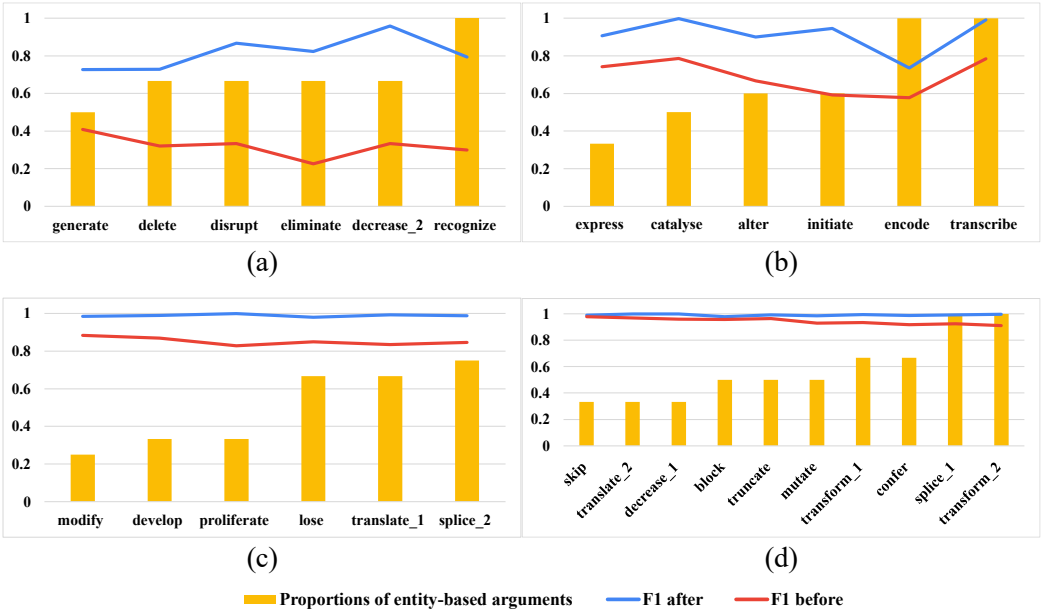


Figure 5. F1 improvements by predicate, grouped by baseline SRL F1 tiers: (a) below 50%, (b) 50% to below 80%, (c) 80% to below 90%, and (d) 90% and above. Within each tier, predicates are sorted by the proportion of entity-based arguments (represented by yellow columns).

or hierarchical typing strategies is also a promissing future approach to mitigate the limitations of argument ambiguity . Leveraging graph-based representations, such as AMR (Abstract Meaning Representation) or UMLS knowledge graphs, could further contextualize arguments and support better predicate-argument alignment.

Moreover, our merging strategy could be extended to integrate additional linguistic features, such as coreference chains or discourse relations, to enable deeper semantic understanding.

Lastly, as although our current focus is biomedical, future efforts may be to improve generalizability beyond domain-specific predicates. This involves applying our model merging method in cross-domain SRL settings to enhance SRL performance in domains with limited annotation.

REFERENCES

[1] W.-C. Chou, R. T.-H. Tsai, Y.-S. Su, W. K., T.-Y. Sung, and W.-L. Hsu, "A semi-automatic method for annotating a biomedical proposition bank," in *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, 2006, pp. 5–12.

[2] M. Miwa, P. Thompson, J. McNaught, D. B. Kell, and S. Ananiadou, "Extracting semantically enriched events from biomedical literature," *BMC Bioinformatics*, vol. 13, no. 1, p. 108, 2012.

[3] P. Thompson, P. Cotter, S. Ananiadou, J. Mcnaught, S. Montemagni, A. Trabucco, and G. Venturi, "Building a bio-event annotated corpus for the acquisition of semantic frames from biomedical corpora," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, 2008.

[4] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep semantic role labeling: What works and what's next," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 473–483.

[5] T. Wattarujeekrit, P. K. Shah, and N. Collier, "Pasbio: Predicate-argument structures for event extraction in molecular biology," *BMC Bioinformatics*, vol. 5, no. 1, p. 155, 2004.

[6] H.-D. Tuan-Nguyen, H.-S. Pham, and V.-T. Hoang, "A semi-automatic approach to biomedical semantic role corpus construction," *Science and Technology Development Journal - Natural Sciences*, vol. 6, no. 2, pp. 2083–2094, 2022.

[7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[8] H.-D. Tuan-Nguyen, L.-T. Tran-Tien-Loi, and V.-H. Le-Dinh, "A deep-learning model for semantic role labelling in medical documents," *Science and Technology Development Journal - Natural Sciences*, vol. 5, no. 2, pp. 1032–1039, 2021.

[9] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics*, 1998, pp. 86–90.

[10] M. S. Palmer and K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," University of Pennsylvania, Computer and Information Science Dept., 2000 South 33rd St., Philadelphia, PA, United States, Tech. Rep., 2005.

[11] S. Pradhan, J. Bonn, S. Myers, K. Conger, T. O'gorman, J. Gung, K. Wright-Bettner, and M. Palmer, "Propbank comes of age—larger, smarter, and more diverse," in *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, 2022, pp. 278–288.

[12] P. Kingsbury and M. Palmer, "From treebank to propbank,"

in *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, M. G. Rodríguez and C. P. S. Araujo, Eds. European Language Resources Association (ELRA), 2002.

[13] O. Majewska, C. Collins, S. Baker, J. Björne, S. W. Brown, A. Korhonen, and M. Palmer, "Bioverbnet: A large semantic-syntactic classification of verbs in biomedicine," *Journal of Biomedical Semantics*, vol. 12, no. 1, 2021.

[14] R. T.-H. Tsai, W.-C. Chou, Y.-C. Lin, C.-L. Sung, W. Ku, T.-Y. Sung, and W.-L. Hsu, "Biosmile: Adapting semantic role labeling for biomedical verbs," in *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, 2006, pp. 57–64.

[15] R. T.-H. Tsai, W.-C. Chou, Y.-S. Su, Y.-C. Lin, C.-L. Sung, H.-J. Dai, I. T.-H. Yeh, W. Ku, T.-Y. Sung, and W.-L. Hsu, "Biosmile: A semantic role labeling system for biomedical verbs using a maximum-entropy model with automatically generated template features," *BMC Bioinformatics*, vol. 8, no. 1, p. 325, 2007.

[16] P.-T. Lai, Y.-Y. Lo, M.-S. Huang, Y.-C. Hsiao, and R. T.-H. Tsai, "Belsmile: A biomedical semantic role labeling approach for extracting biological expression language from text," *Database*, 2016.

[17] R. T.-H. Tsai and P.-T. Lai, "A resource-saving collective approach to biomedical semantic role labeling," *BMC Bioinformatics*, vol. 15, no. 1, pp. 160–171, 2014.

[18] F. Eckert and M. Neves, "Semantic role labeling tools for biomedical question answering: a study of selected tools on the bioasq datasets," in *Proceedings of the 6th BioASQ Workshop A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering*, 2018, pp. 11–21.

[19] A. D. P. Ariyanto, D. Purwitasari, and C. Fatichah, "A systematic review on semantic role labeling for information extraction in low-resource data," *IEEE Access*, vol. 12, pp. 57 917–57 946, 2024.

[20] F. Ikhwantri, S. Louvan, K. Kurniawan, B. Abisena, V. Rachman, A. F. Wicaksono, and R. Mahendra, "Multi-task active learning for neural semantic role labeling on low resource conversational corpus," in *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, 2018, pp. 43–50.

[21] H.-D. Tuan-Nguyen, T. Duong Luu, and Q. Duy Huynh, "A syntax-aware deep-learning model for biomedical semantic role labelling," *Science and Technology Development Journal - Natural Sciences*, vol. 7, no. 4, pp. 2750–2762, 2023.

[22] W. Liu, S. Yang, and K. Tu, "Structured mean-field variational inference for higher-order span-based semantic role," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 918–931.

[23] S. S. S. Das, H. Zhang, P. Shi, W. Yin, and R. Zhang, "Unified low-resource sequence labeling by sample-aware dynamic sparse finetuning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 6998–7010.

[24] J. G. Moreno, J. Lovon-Melgarejo, M. Rick, Robin-Charlet, C. Damase-Michel, and L. Tamine, "Patientdx: Merging large language models for protecting data-privacy in healthcare," in *Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health)*. Albuquerque, New Mexico: Association for Computational Linguistics, May 2025, pp. 1–11, retrieved June 1, 2025. [Online]. Available: https://aclanthology.org/2025.cl4health-1.1/

[25] T. Rousset, T. Kakibuchi, Y. Sasaki, and Y. Nomura, "Merging language and domain specific models: The impact on technical vocabulary acquisition," February 2025, preprint on arXiv.

[26] J.-P. Corbeil, A. Dada, J.-M. Attendu, A. B. Abacha, A. Sordoni, L. Caccia, F. Beaulieu, T. Lin, J. Kleesiek, and P. Vozila, "A modular approach for clinical slms driven by synthetic data with pre-instruction tuning, model merging, and clinical-tasks alignment," May 2025, preprint on arXiv.

[27] Y.-A. Lee, C.-Y. Ko, T. Pedapati, I.-H. Chung, M.-Y. Yeh, and P.-Y. Chen, "Star: Spectral truncation and rescale for model merging," in *Proceedings of the 2025 Conference of the North, Central and South American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Albuquerque, New Mexico: Association for Computational Linguistics, April 2025, pp. 496–505, retrieved June 1, 2025. [Online]. Available: https://aclanthology.org/2025.naacl-short.42/

[28] T. Parekh, I.-H. Hsu, K.-H. Huang, K.-W. Chang, and N. Peng, "Geneva: Benchmarking generalizability for event argument extraction with hundreds of event types and argument roles," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 3664–3686.

[29] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 1, pp. 1–23, January 2022. [Online]. Available: https://doi.org/10.1145/3458754

[30] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," April 2019, preprint.

[31] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," in *Proceedings of the Eleventh International Conference on Learning Representations*, February 2023.

[32] E. Yang, Z. Wang, L. Shen, S. Liu, G. Guo, X. Wang, and D. Tao, "Adamerging: Adaptive model merging for multi-task learning," in *Proceedings of the International Conference on Learning Representations (ICLR 2024)*, October 2024. [Online]. Available: https://arxiv.org/abs/2310.02575

[33] M.-S. Huang, P.-T. Lai, R. T.-H. Tsai, and W.-L. Hsu, "Revised jnlpba corpus: A revised version of biomedical ner corpus for relation extraction task," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, 2024, pp. 73–78.

Hoai-Duc Tuan-Nguyen earned his B.Sc. in Information Systems in 2005 and his M.Sc. in Computer Science in 2009 from the same university, where he is currently pursuing a Ph.D. Now he is a lecturer in the Faculty of Information Technology at the University of Science, Vietnam National University, Ho Chi Minh City. His research interests include deep learning, natural language processing, and explainable artificial intelligence.

Email: tnhduc@fit.hcmus.edu.vn

An Approach for Dealing with Sequential Data in Intelligent Tutoring Systems

Nguyen Xuan Ha Giang[1,2], Lam Thanh-Toan[2], Nguyen Thai-Nghe[1*]
[1]Cantho University, Cantho City, Vietnam
[2]Cantho University of Technology, Cantho City, Vietnam
Correspondence: Nguyen Thai-Nghe. Email: ntnghe@cit.ctu.edu.vn
Communication: received 23/02/2025, revised 24/06/2025, accepted 04/07/2025
Digital Object Identifier: 10.32913/mic-ict-research.v2025.n2.1355

Abstract: The use of educational data to gain deeper insights into learners' interaction histories with Intelligent Tutoring Systems (ITS) is receiving increasing attention, especially in the context of online learning and the growing demand for digital transformation in education. Predicting learners' academic performance through the analysis and evaluation of their recorded activities in ITS plays a critical role in supporting educational administrators and instructors. An understanding of learners' abilities helps refine teaching methods and optimize learning environments, ultimately enhancing educational quality. Our research improves upon our previous work, which utilized only LSTM, by incorporating an ensemble model - CL-PSP, combining CNN and LSTM networks. Specifically, the study focuses on predicting learners' CFA capability - the likelihood of learners answering correctly on their first attempt. Knowledge evolves over time, observed in users' interaction preferences within session-based recommendation systems. CL-PSP leverages critical factor in shaping learners' academic performance in two educational datasets, KDD Cup 2010 and Assistment 2017. Several enhancements, including a revised ensemble architecture, improved error measurement, and refinements in data preprocessing. Results demonstrate that the proposed model significantly outperforms existing models, achieving superior performance with a lower Root Mean Square Error (RMSE). On the KDD Cup 2010 dataset, the model achieves a minimum RMSE of 0.375, while notable improvements are also observed on the Assistment 2017 dataset, further underscoring the model's effectiveness and robustness. The experimental results underscore the feasibility and considerable potential of utilizing session-based data in ITS to enhance both learning performance and educational quality.

Keywords: *Ensemble CNN-LSTM, Intelligent Tutoring Systems; Performance prediction; Session-based recommender system.*

I. INTRODUCTION

Intelligent Tutoring Systems are computer-based teaching and learning management systems that provide personalized assessments of individual competencies [1]. Within their operational architecture, ITSs enhance interactive environments, learning experiences for both learners and instructors through the integration of artificial intelligence techniques, including: *(1) Machine learning techniques*: encompass methods such as classification, clustering, regression, and reinforcement learning, which are utilized to predict learners' abilities based on current interaction data or learning history. These techniques enable the dynamic adjustment of the learning environment, facilitating personalized learning improvements through predictive modeling. *(2) Data mining techniques*: Data mining techniques: involve the processing, analysis, searching, and extraction of features using various analytical methods, such as clustering, multivariate analysis, and quantitative approaches, to derive insights from educational data. *(3) Recommendation techniques*: offer recommendations for relevant knowledge, course materials, problem-solving strategies, resources, and exercises tailored to the learner's progress and proficiency. *(4) Natural language processing (NLP) techniques*: are employed for syntactic and semantic analysis to comprehend, classify, and evaluate the feedback and needs of both learners and instructors. *(5) Expert knowledge*: as advisory tools, providing context-specific solutions and recommendations for both learners and instructors, based on individual circumstances and learning needs.

Recommender systems (RS) have emerged as a vital tool for processing end-user data. Particularly in addressing the challenges posed by vast volumes of information, diverse formats, and complex structures across various domains. Advances in RS techniques have revolutionized data utilization, enhancing efficiency in areas such as e-commerce, entertainment, and essential fields like education and training. One of the most popular approaches is collaborative filtering (CF), which generates recommendations based on

the behaviors and preferences of similar user groups from the past. This method assumes that users are likely to enjoy content favored by others with similar preferences. CF is typically categorized into two main types: *(1) User-based CF*: analyzes users' rating histories to identify "similar" users (measures such as Cosine, Euclidean, or Pearson). It then recommends items that these similar users have liked but the current user has not yet explored. *(2) Item-based CF*: evaluates the similarity between items based on how they have been rated or used by users. It then suggests items similar to those the user has previously liked or rated highly. Beyond CF, other techniques commonly employed in RS. Content-based filtering: recommendations are generated by analyzing the attributes of items. Knowledge-based filtering: this approach leverages domain-specific information and rules tailored to user needs or contexts. Hybrid methods: these combine multiple approaches to enhance accuracy and relevance. Ultimately, to optimize RS performance and enhance user experiences, the flexibility to integrate and adapt various methods remains a key factor across diverse applications.

In the context of ITS, session-based recommendation systems (SBRS) play a crucial role in providing accurate and timely learning suggestions. Instead of relying on past learning history, SBRS leverage students' behavior and interactions during each session. This enables personalized recommendations, thereby optimizing the learning experience. This approach is particularly important as students' learning objectives can change rapidly throughout the learning process. Numerous studies have focused on early prediction of learning outcomes based on data collected from ITSs. A study by Li [2] applied a 1D CNN (Convolutional Neural Networks) model to extract data features, combined with LTMS, to predict the course outcomes of students preparing for the start of a term. The experimental results were evaluated with an RMSE metric of 0.785. Further research by [3, 4] expanded this model by utilizing deep convolutional layers and bidirectional LSTM (Long Short-Term Memory networks) models to forecast complex phenomena such as gold price fluctuations and climate change over time. Additionally, Pan et al. [5] developed a diverse course RS that employs deep learning algorithms to extract features from educational data, thereby optimizing the curriculum for individual learners. A currently prominent research direction is the prediction of student performance particularly their ability to correctly respond on the first attempt to exercises posed by the ITS. This prediction not only helps assess the readiness of the learner but serves as the basis for optimizing teaching systems and assessments. The aim of designing curricula appropriately meets the specific needs of each student and enhance teaching effectiveness.

CFA is a key metric for assessing learners' knowledge readiness in ITS. This metric reflects the learner's ability to correctly answer a question or exercise on their first attempt. It also provides insight into their understanding and preparation for the learning content. CFA is widely used to predict learning outcomes in automated educational systems. It is integrated into ITS and plays a vital role in several areas. *(1) Assessment of learning ability*: Assessment of learning ability: CFA helps predict the likelihood of a learner answering a question correctly on their first attempt. It also forecasts overall learning performance throughout the course. Based on this information, the ITS can adjust question difficulty, ensuring an appropriate level of challenge that supports the learner's improvement. *(2) Personalization of learning progress and immediate feedback*: CFA allows the personalization of the learning path for each learner based on their CFA score. The system can provide advanced exercises for those with high CFA scores. For those with lower CFA scores, it can offer remedial lessons with instant feedback to help those with lower CFA scores improve. *(3) Adjustment of teaching strategies*: CFA enables the ITS to automatically recognize and adjust its teaching strategies. It does so by suggesting relevant lessons or materials that support the learner's progress, ensuring that the learning path is aligned with their abilities. *(4) Data analysis*: The effectiveness of lesson design, exercises, and questions is evaluated through CFA. A low CFA value may indicate that the content is too challenging, not aligned with the required knowledge, or lacks sufficient grounding. This can lead to gaps in the learner's problem-solving ability.

Building on our previous work presented at the FDSE 2023 conference [6], we introduce the ensemble CL-PSP model, which combines CNN and LSTM. This model is designed to predict learners' performance by analyzing their ability to correctly answer questions posed by ITS. To evaluate its effectiveness, we utilized two widely recognized educational datasets: KDDCup 2010 and Assistment 2017. The key contributions of this study include:

(1) Introduction of the CL-PSP model: We propose CL-PSP, a hybrid architecture that integrates static feature extraction within learning sessions using CNN and captures temporal dynamics through LSTM networks. This combination allows CL-PSP to accurately predict the CFA, an important indicator of a learner's ability to answer correctly on the first try. CFA has significant practical implications for educators and educational administrators, as it supports more effective planning, design, and adaptation of instructional activities.

(2) Demonstration of CL-PSP's superior performance: On the KDDCup 2010 dataset, the CL-PSP model achieved

a 2.60% reduction in RMSE compared to TAGNN_CFA. Although the results on Assistment 2017 were less pronounced, the RMSE of CL-PSP was approximately 5.47% higher than that of TAGNN_CFA, indicating potential for improvement. Additionally, our previously published LSTM-based model showed slightly inferior performance, with an RMSE increase of about 15.20% and 0.63% compared to CL-PSP, respectively. These results confirm the effectiveness and practical potential of CL-PSP in ITSs, particularly for tasks involving learning performance prediction through comprehensive comparative analysis.

II. RELATED RESEARCH AND RECOMMENDATION SYSTEMS

1. Related works in predicting academic performance

In the context of modern education, data mining has become an essential tool for enhancing learning efficiency and improving educational quality. Numerous studies have applied advanced approaches such as machine learning, neural networks, and hybrid models to predict learning performance, analyze behavior, and optimize learning processes. This study categorizes and highlights notable research according to these approaches, thereby emphasizing their practical applications in the educational domain.

(1) Machine learning approaches: Classical methods such as Decision Trees, KNN, SVM, XGBoost, LightGBM, Random Forest, and ensemble techniques have been widely used to predict CFA and learning performance from static interaction data in ITS and LMS platforms. BKT [7] and MF [8] proved effective in capturing knowledge accumulation. To improve classification, Tariq et al.[9] tested oversampling techniques, with KNN achieving the best result (83.7%) on a multi-class LMS dataset. While [10] combined SVM and ANN with an optimization method to predict pass/fail outcomes and final scores. These approaches highlight the role of preprocessing and feature selection in boosting performance on static data. The research [11] used Logistic Regression, Random Forest, and Naive Bayes to predict graduation GPA from semester-wise data of 357 students. Accuracy, precision, and F1-score improved in later semesters (up to 85%) as learning data accumulated, while early predictions were less reliable due to limited evidence. Though effective for cumulative prediction, the approach oversimplifies temporal variation and early learning progression. Hoq et al.[12] introduced an explainable ensemble model using KNN, SVM, and XGBoost, achieving RMSE = 0.151 on programming data. While SHAP enhanced interpretability, these models lack sequential modeling. Our CL-PSP combines sophisticated deep learning techniques to predict continuous learning behaviors and analyze time-series data, offering the ability to personalize predictions for individual learners.

(2) Neural network approaches: Recent deep learning approaches have explored various architectures to model student behavior and enhance predictive accuracy. STR-SA [13] applied an attention-based network to recommend discussion threads in MOOCs, capturing session-level interactions but lacking cumulative learning context. ClickTree [14] combined BiGRU with a tree-based decoder to model short-term clickstream behavior while producing interpretable predictions (AUC = 0.788), yet it focused only on within-session data. Wang et al. [15] extended the Deep Knowledge Tracing framework by incorporating fatigue-related features-such as session duration and activity intervals-into an RNN-based model to capture cognitive state transitions. While it improves intra-session prediction, the model relies heavily on accurate timestamp features and lacks multi-session generalization. DSMKT [16] introduced a dual-branch architecture combining masked self-attention and GRU, further enhanced with online knowledge distillation. It achieved strong results on multiple datasets (AUC = 0.8378, ACC = 0.7827 on ASSISTments09), though it still focuses primarily on next-step prediction without long-range learner modeling. GGCNN [17] employed Graph Convolutional Networks with Genetic Algorithms to model structured student-course relationships, achieving up to 98% accuracy and F1 = 0.84. Meanwhile, MixerKT [18] used a pure MLP-based architecture to model learning sequences efficiently, reaching AUC = 0.7575 and KC accuracy = 0.7171 on AL2005. However, both approaches lack sequential personalization over time. While each of these models contributes uniquely-whether through attention, RNNs, GCNs, or MLP-based designs-they often fail to capture long-term behavioral trends across sessions. Our proposed CL-PSP model addresses these limitations by integrating CNN and LSTM to jointly learn local patterns and long-range temporal dependencies, enabling personalized and time-aware CFA prediction.

(3) Hybrid models: Hybrid architectures combine complementary neural components to model complex educational and prediction tasks. *(i) Temporal–structural modeling*: Several studies incorporate graph-based representations into sequential learning frameworks. CLGT[19], a graph transformer model that leverages heterogeneous student interaction graphs and temporal position embeddings for predicting performance in collaborative learning. CLGT outperformed GCN, GAT, and MLP on two real-world datasets, but its effectiveness depends heavily on graph quality and scenario-specific generalization. GC-LSTM[20] predicted dynamic links by feeding graph snapshots into an LSTM, effectively capturing temporal dependencies

and structural transitions. [21] applied a similar model to traffic forecasting, reporting reduced MAE and RMSE. However, these models often face challenges when applied to sparse graphs or rapidly evolving network structures. *(ii) Temporal–spatial modeling*: CNN-LSTM hybrids are widely used in time-series tasks with both spatial and temporal features. CNN-LSTM model applied by [22] successfully predicted indoor temperature spikes, achieving a MAPE between 0.89% and 5.46%. Alhussein et al. [23] used the same architecture for household load forecasting with high short-term accuracy. [24] used the same architecture for household load forecasting with high short-term accuracy. In education, [24] leveraged CNN-LSTM to extract key performance indicators from LMS interaction logs, outperforming models. A recent study by Alnasyan et al. [25] benchmarked several deep learning architectures—including CNN, RNN, LSTM, Bi-LSTM, and GRU-on student performance prediction tasks using LMS activity logs. Among all models, BiLSTM achieved the highest performance with accuracy of 90%, outperforming CNN by 4.7%. The study highlight the importance of bidirectional temporal modeling in capturing behavioral patterns from student log data. Despite strong performance, these models often require high computational resources and struggle with large-scale or noisy data. Deeper architectures are also prone to overfitting when applied to imbalanced datasets. *(ii) Temporal–semantic modeling*: To capture semantic context in educational data, LBKT [26] integrated BERT embeddings with LSTM and Rasch-based encoding, improving AUC and memory efficiency on long sequences. Building on this direction, [27] proposed MCQStudentBERT, a transformer-based model that incorporates student history and question-answer semantics to predict answer choices. Their concatenation-based variant achieved (0.579) and F1 score (0.740) using Mistral 7B embeddings, outperforming summation-based approaches. [28] explored large language models for learner performance prediction by fine-tuning GPT-2 on sequences of question content, options, and correctness. On the ASSISTments 2015 dataset, GPT-2 (medium) achieved an AUC (0.83) and accuracy (81%), outperforming DeepIRT, a deep neural variant of item response theory, and DKVMN, a memory-augmented model for multi-skill tracing. TAGNN-Lazy[29], an extension of the original TAGNN model[30], improves performance in session-based product recommendation by using directed graphs and GNNs with attention to capture key user interactions. Evaluated on RSC15 and Diginetica, it outperforms baseline models in Precision@20 and MRR@20.

While effective in modeling long-range semantic and temporal patterns, these models incur high computational costs, limited generalizability, and reduced interpretability. Although each hybrid strategy targets specific aspects, challenges remain: structural models struggle with sparse and dynamic graphs; spatial models are resource-intensive and sensitive to noisy data; semantic models trade scalability and transparency for expressiveness. To address these limitations, the proposed CL-PSP integrates CNN for local feature extraction and LSTM for temporal dynamics, enhancing CFA prediction in ITSs.

2. Models in recommendation systems

LSTM is a type of RNN specifically designed to process and predict sequential or time-series data. The strength of LSTM lies in its ability to retain information over extended time periods. It addresses the issue of long-term dependencies and overcoming the vanishing gradient problem typically encountered in traditional RNNs [31]. Each LSTM layer is capable of preserving information from previous computations through its unique structure. An LSTM layer (Figure 1) comprises multiple cells, each containing three essential gates [32]. These gates are critical for controlling the flow of information through the network over time. They act as mechanisms to open or close pathways. This enables the network to decide whether to retain, add, or discard information in the memory state. This is achieved through the sigmoid activation function and matrix multiplication operations. LSTM networks include three primary gates. (1) Forget Gate (f_t): this gate determines which portions of information from the previous hidden state (h_{t-1}) should be discarded. Here, (x_t) represents the current input, and σ denotes the sigmoid activation function. (2) Input Gate (i_t): this gate decides which new information will be added to the memory state. It also updates the candidate memory state $\widehat{C_t}$. (3) Output Gate o_t: this gate determines which parts of the current hidden state h_t should be used to generate the output. It also updates the current memory state C_t.

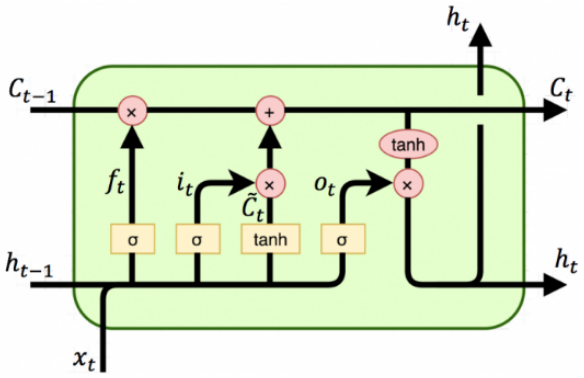


Figure 1. LSTM memory structure.

The proposed CL-PSP model is implemented with three LSTM layers, measuring the discrepancy between the actual labels and the model's predictions in binary classification tasks using the binary_crossentropy loss function (Eq.1.). This function optimizes the model by calculating the loss, the difference between the true labels (y) and the predicted labels ($\hat{y_i}$), to minimize the loss and improve accuracy. Here, (y) represents the ground truth label (a value of 0 or 1), and ($\hat{y_i}$) denotes predicted probability, typically obtained through the sigmoid activation function. The function [Eq.2.] transforms the output (s) into a probability value within the range [0,1].

$$Loss = -y_i log(\hat{y_i}) - (1 - y_i) log(1 - \hat{y_i}) \quad (1)$$

$$\hat{y_i} = \frac{1}{1 + e^{-s_i}} \quad (2)$$

Adam (Adaptive Moment Estimation) [33] is an optimization algorithm designed for rapid convergence by dynamically adjusting the learning rate for each parameter based on gradient information. It ensures a stable training process, particularly for sequential or complex time-series data. Adam helps fine-tune the weights during training to effectively minimize the loss function.

CF is a technique that can be used to predict learning outcomes, recommend appropriate learning methods, and analyze the progress of learners based on data from students with similar learning behaviors. According to Breese et al. [34], CF can be categorized into two main approaches: Model-based and Memory-based. *(1) Model-based method*: this approach employs machine learning algorithms to create prediction models using user and product data. For instance, techniques like MF or Singular Value Decomposition can be applied to uncover hidden relationships between factors that influence learning ability. These factors include test results, study habits, and classroom participation. This model can predict a student's scores on upcoming tests based on similarities with other students' learning characteristics. It effectively handles sparse data, scales well, and uncovers latent relationships within the data. *(2) Memory-based method*: this is the traditional approach in CF. Predictions about user preferences or behaviors are made by directly calculating similarities between users or products based on historical behavior data, without the need for machine learning models. Memory-based CF includes two main types: User-based CF and Item-based CF.

According to [35], CF considers the values users assign to products as linked features, forming the basis for filtering large datasets and supporting predictions and recommendations. CF is useful in three key tasks. (1) Predicting learning outcomes: Estimating students' scores or academic performance based on the learning behaviors of students with similar preferences or results. (2) Recommending learning methods: Suggesting study methods, learning materials, or lessons based on the study habits and performance of similar students. (3) Analyzing and predicting learning progress: Forecasting students' future progress or changes in performance, based on past data and the learning behaviors of other learners. Fundamental formulas in CF include: *(1) User Mean* (Eq.3.): the average value of the user rating scores assigned to the product is calculated. The average helps to balance the bias in user ratings, particularly when there are significant differences in rating levels. For $Rate_{u,i}$: the rating given by user u to item i; $Item_u$: the set of items rated by user u. $\overline{Mean_u}$: the average rating of u.

(2) Pearson Correlation [36]: it measures the similarity between two users based on their ratings of common items. It determines the degree of association between users through their ratings of similar products (Eq. 4). Here, $i \in Item_{u,v}$ refers to the set of items that both u and v have rated; $Sim_{(u,v)}$ represents the similarity between users u and v. *(3) Prediction of Ratings* (Eq.5): the prediction estimates the score a user will assign to an item based on the similarity between users or items. This estimate is computed by aggregating the ratings of similar users, adjusted to account for variations in users' average ratings. Let $Pred_{u,i}$: denote the predicted rating of u on i; N refers to the set of neighbors v of u for i. The value of the adjustment coefficient k can be identified through experimental trials and performance evaluation, where various values are assessed to optimize the model's accuracy and prediction quality.

$$\overline{Mean_u} = \frac{1}{|Item_u|} \sum_{i \in Item_u} Rate_{u,i} \quad (3)$$

$$Sim_{(u,v)} = \frac{\sum_{i \in I_{uv}} (Rate_u - \overline{Mean_u})(Rate_v - \overline{Mean_v})}{\sqrt{\sum_{i \in I_{uv}} (Rate_u - \overline{Mean_u})^2 \cdot \sum_{i \in I_{uv}} (Rate_v - \overline{Mean_v})^2}} \quad (4)$$

$$Pred_{u,i} = \overline{Mean_u} + k \sum_{v=1}^{N} Sim_{(u,v)} \left(Rate_{v,i} - \overline{Mean_v} \right) \quad (5)$$

In our study, the User-based CF technique was employed, utilizing user similarity computed through the K-Nearest Neighbors (KNN) algorithm. This method was applied to estimate correlations and predict students' ability to answer questions.

MF: when approached mathematically, involves decomposing a large data matrix R into the product of two or more smaller matrices, P and Q. Each of these matrices,

has reduced dimensions, achieved through a lower−rank representation that retains the essential information. This enables the identification of hidden patterns or features within the data. This method is widely used in various fields such as recommender systems, prediction, and deep learning. The result of this factorization process is the connection of sub−matrices. The goal is to reconstruct the original matrix R from P and Q while minimizing the error, ensuring that R is more accurate [37]. Key operations include the Loss Function, which measures measures the discrepancy between the actual value r_{uj} and the predicted value $\hat{r}_{ui}$ (Eq.6). It also includes the optimization function based on Frobenius norm (Eq.7). Under the assumption that $R[m,n]$ represents the input matrix and $P[m,k]$, $Q[k,n]$ are the optimal sub−matrices. MF in RS utilizes stochastic gradient descent (SGD) to predict missing values in the rating matrix, thereby optimizing the objective function.

Algorithm 1 KNN Algorithm for Performance Student Prediction

Require: X_{train}: User–item rating matrix $[u,i]$; X_{value}: Values of training data $[0,1]$; Y_{pre}: User–item prediction vector;

N: User-neighbors list on item i; K: Number of nearest neighbors.

Ensure: $\hat{y}_{\text{pre}}$: Predicted value of user Y_{pre}.

Step 1: Initialization

$distance \leftarrow \emptyset$ ▷ A distance list between u and its neighbors

Step 2: Main Process

for $t = 1$ to $|X_{\text{train}}|$ **do**

(1) Computing distance

$d[i] \leftarrow similarity(X_{\text{train}}, Y_{\text{pre}})[i]$

$distance[i] \leftarrow X_{\text{value}}[i]$

(2) Sorting

Sort $distance$ in ascending order

$N \leftarrow$ first K elements of $distance$

(3) Prediction

$rating \leftarrow$ average distance of $N[i]$

$\hat{y}_{\text{pre}} \leftarrow$ highest $rating$ in N

end for

return $\hat{y}_{\text{pre}}$

$$\hat{r}_{ui} = P_u.Q_i^T = \sum_{k=1}^{K} P_{uk}.Q_{ik} \tag{6}$$

$$L = \frac{1}{2} \sum_{(u,i)\in R} (r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda}{2}(\|P\|_F^2 + \|Q\|_F^2) \tag{7}$$

SGD operates as follows: Random initialization of P, Q. (2) Updating each vector p_i, q_j (rows of P, Q respectively) based on the partial derivatives of the gradient to reduce the slope of the prediction error relative to the actual values. The updating process (Eq.8, 9) continues until the objective function L converges, reaching its minimum value. Let η (0¡ η ¡1) denotes the learning rate.

$$p_i \leftarrow p_i + \eta.[2.(r_{ui} - \hat{r}_{ui})^2.q_j - 2\lambda p_i] \tag{8}$$

$$q_j \leftarrow q_j + \eta.[2.(r_{ui} - \hat{r}_{ui})^2.p_i - 2\lambda q_j] \tag{9}$$

CNNs [38]: stand as a highly efficient deep learning architecture widely employed in diverse domains, including computer vision and speech recognition. Through the implementation of convolution operations at diverse levels of granularity, CNNs demonstrate exceptional proficiency in extracting essential non-linear features for learning tasks. This characteristic substantially diminishes the dependence on manual feature engineering. As a result, CNNs are particularly well-suited for deployment in predictive systems and SBRS. LeNet-5 is a prevalent CNN architecture, comprising layers: Convolutional, pooling, and fully-connected layers.

(1) Convolutional layers: consist of input data channels (I). The feature maps (O) contain output matrices of neurons. Each neuron is derived from the application of a filter to a region of the input data in the preceding convolutional layer. The filters K are the convolution kernels with x, y representing the indices of the row and column. The calculation of the value for a recently introduced neuron, positioned at coordinates (i, j) within the feature map associated with the k^{th} filter, is determined by (Eq.10). This value results from convolution with the c^{th} input channel. H, V represent the horizontal and vertical directions of the 2D convolution kernel, while x, y denote the indices of the corresponding row and column in this kernel for channel c. Let $K[p,q,c]$ signify the weight of the neuron at position p, q in the k^{th} filter. And $I(i+p, j+q, c)$ is the neuron value at the position $i+p, j+q$ in the preceding input layer (c^{th} channel). Here, b is the bias of the c^{th} input. Denote (f) as the $Sigmoid$, $Tanh$, or $ReLU$ activation function, depending on the CNN model architecture.

$$O_{[i,j,c]} = f(\sum_{p=0}^{K_{x-1}} \sum_{q=0}^{K_{y-1}} (I_{[i+p,j+q,c]}.K_{[p,q,c]}) + B_{[c]}) \tag{10}$$

(2) Pooling layers: also known as subsampling layers, are usually positioned immediately after Convolutional layers. They decrease the number of neurons, diminish the size of subsequent feature maps, and preserve vital information

from the features. Two widely used subsampling techniques are Max Pooling and Average Pooling. In essence, the neurons in each feature map of these Pooling layers are calculated from and connected to the preceding convolutional layers. And (3) Fully-connected layers or dense layers: Each neuron within this layer is an combination of all neurons from the previous layer. The aim is to produce a comprehensive semantic information layer, encompassing essential predictions or classifications.

III. DATA DESCRIPTION

1. Similarities between SBRS and learner assessment problem

KDDCup 2010 and Assistment 2017 were the two datasets utilized in our experiments, sourced from different ITS. Each dataset comprises a large collection of records in numerical or textual format. Each record contains multiple data fields representing corresponding feature values. Each record represents a question answered by a learner, including whether the answer was correct or incorrect. It also contains other critical information such as problem, learner, time, response duration, and knowledge component (KC) related to the question (Table I). During the question-solving process, learners could request assistance or knowledge hints provided by the ITS. Finally, the learner's response concludes the session for the current question.

Table I
OVERVIEW OF THE DATASET EXTRACTED FROM ITSS

Row	1	...	n
User	User 1	...	User n
Problem	Problem 1	...	Problem m
Step	Step 1	...	Step 2
CFA	0	...	0
KC	KC1	...	KCn
Start time	Value	...	Value
...	...	...	...

According to [39], a session is an independent data unit, clearly defined by a start and end timestamp, and may or may not follow a sequential order. The sequentiality in the two datasets is captured through the chronological order of learners' interactions with the system, during answering process. In these datasets, each session is represented in a data entry format, reflecting an independent, sequential transaction. Each session includes information such as the learner's details, the problem, relevant knowledge components, timestamps, question responses, and other related attributes. Due to the nature of the datasets, learners may appear intermittently in additional exercises or across different session periods. To improve the accuracy of predictions

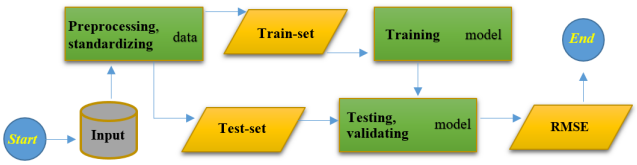


Figure 2. The approach to training and testing.

for the current step, we have organized the data sequentially based on the timing of each learner's interactions.

2. Data splitting method

In this educational setting, the main objective is to predict learning performance by forecasting the CFA value at the current step. This helps learners in mastering the necessary KCs. Our research experiments are focused on developing session-based data processing models to achieve this goal. The key features used in the training process include: (1) Student identifiers: learner identification. (2) Assignment identifiers: an exercise containing many questions. (3) Answer indicators: *CFA*, correctindicating whether the answer to the selected question is true or false. (4) Timestamps: start time, end time. Each row represents a single step, corresponding to a session, a response.

After preprocessing and normalization, the raw data undergoes feature extraction (Figure 2). To account for the temporal sequence, the datasets are organized by learners and the order of their question responses. This order serves as the grouping criterion. Each cohort of users is then split into two parts, with 70% assigned to the first segment and 30% to the last segment. These segments are aggregated to create the training set, comprising 70% of the source data. The remaining 30% is reserved for the testing set (Figure 3). The target value to be predicted is binary, where 1 indicates a correct response from the learner and 0 denotes an incorrect response to the respective question. The difference between the predicted and actual values is evaluated using the RMSE metric. The error evaluation process on the testing sets does not include the CFA field.

Each assignment will contain many independent and interrelated steps so that the final result can be obtained. Instructional items are related to questions and steps involved in problem-solving. An example of a problem: *"Calculate the radius of the circle with center O and radius R¨*. This is considered an exercise, where *"calculate the radius R"* becomes a question, a step. When the learner answers this question, the information is recorded as a session. Table II presents the general structure of the two training datasets and the features used to train our CL-PSP model. The KDDCup 2010 comprises 23 columns and 93,195 rows. Data from 70 learners who answered 6,666 questions

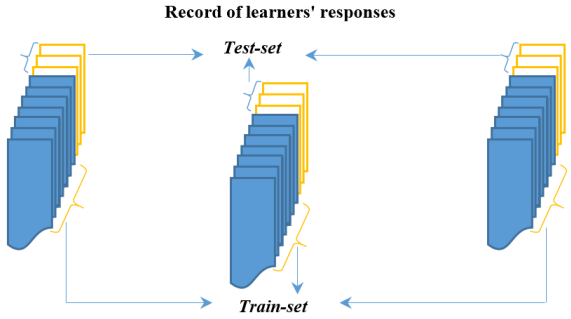


Figure 3. An overview of the train-set and the test-set [40].

were used across a total of 59,864 training sessions and 33,331 test sessions. In contrast, the Assistment 2017, obtained from 276 learners who solved 204 questions. It consists of a total of 231,403 rows and 76 columns. The corresponding training and testing sets contain 154,204 and 77,199 sessions, respectively.

Table II
DATASET ARCHITECTURE

Dataset		KDDCup 2010	Assistment 2017
Session count	Train-set	59,864	154,204
	Test-set	33,331	77,199
Features		23	76
Training features		5	5
Problems		6,666	204
Users		70	276
Average No. steps/learner	Train-set	855	559
	Test-set	497	424

Given the nature of the system, each learner engages differently when answering questions and exercises. Therefore, we estimate the average number of questions answered by each learner on the ITS. This helps support data observation and statistical analysis in the following section. This experimental version is based on the observations of characteristics recorded by ITS. Specifically, the number of interactions per user varies across the datasets. In KDDCup 2010, users are required to provide responses without the option to skip, whereas Assistment 2017 does not impose such a constraint. Consequently, we selected and processed data only for users with interaction counts ranging from 2 to 20 for the KDDCup 2010 dataset and from 2 to 50 for the Assistment 2017 dataset.

IV. EXPERIMENTAL DESIGN

To evaluate the effectiveness of the CFA prediction models, experiments were conducted on the two widely-used educational datasets. The models under comparison include User Average, CF, MF, LSTM, TAGNN_CFA and the proposed hybrid model CL-PSP. For fairness, each model was trained using carefully selected and optimized hyperparameters.

1. Data preparation

The student and problem identifiers were encoded using LabelEncoder, while one-hot encoding was applied to convert them into discrete integer indices or feature vectors, depending on the model used for training. The interaction sequences for each student were sorted chronologically. The dataset was split into 70% for training and 30% for testing (in Table II)

2. Baseline models and hyperparameters

The User Average model predicts the CFA of a student by calculating the average of their previous CFA values. This serves as a simple baseline for comparison with more complex models. CF, model adopts a user-based approach, where similarity between students is computed using the Pearson correlation coefficient. CFA is predicted based on a weighted average of the responses from the top $K = 20$ most similar students. The model input includes learner-problem matrix and uses historical CFA values to infer predictions. MF is implemented by learning two latent matrices representing users and items, respectively. The hyperparameters include as follow: latent dimension $K = 2$, regularization coefficient $\lambda = 0.1$, learning rate = 0.005, and and number of training epochs = 25. The model is optimized using gradient descent by minimizing the squared error between the predicted and actual values.

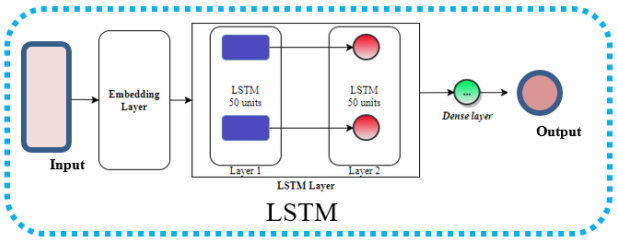


Figure 4. Prediction model from our previous work (FDSE 2023, [6]).

The LSTM model in Figure 4 is based on the architecture introduced in our previous study. It consists of two layers, each with 50 units, and a fully connected layer. This is implemented using the Keras library with the Sigmoid activation function. The model is trained using the Adam optimization algorithm, with a batch size of 1 and a learning rate of 0.0001. An early stopping mechanism is applied if the loss does not improve over 5 consecutive epochs. This

architecture has shown stable prediction results across both datasets, with low error and good generalization ability. TAGNN[30] is used as a baseline, adapted for CFA prediction by reformulating it as a binary classification model. TAGNN_CFA represents each session as a directed graph of learner interactions, with attention mechanisms highlighting the most relevant past actions. The model is implemented in PyTorch, trained with BCEWithLogitsLoss, and optimized using Adam. Hyperparameters include hidden dimensions (64-128), batch sizes (64–128), and learning rates (0.001-0.0005), with early stopping to prevent overfitting.

Building on this, our study proposes a hybrid CNN-LSTM model (CL-PSP) for predicting learner performance. CL-PSP combines the strengths of artificial neural networks to predict learners' academic performance. It does so across two distinct session-based datasets in the educational domain.

3. Model proposal

Figure 5 delineates the structure of CL-PSP, which integrates co-occurring single models in a time-sequential manner. The model targets the prediction of CFA values by leveraging temporal features, particularly the learner's question response start time and end time. Let t represent the current time step for prediction. While $(t-1)$, contextual features such as learner ID, exercise, number of steps, start time, and end time refer to the sequence of previous responses. These temporal signals are encoded and provided as inputs to both CNN and LSTM modules. The main objective is to use the time-based progression to determine whether the learner will answer the current question correctly. Training was conducted on the datasets with a maximum of 10 and 50 epochs, based on early stopping as no significant improvements were observed beyond these points. Hyperparameters were selected using RMSE as the primary evaluation metric. (1) The LSTM module is composed of three layers, each with 50 hidden units. The first two layers are configured with return_sequences=True to maintain temporal continuity, while the final layer outputs a tensor of size (None, 50). Dropout (0.5) is applied to reduce overfitting, and the model is trained using the Adam optimizer with a learning rate (0.001) and a batch size (32). LSTM layers {1,2,3,4} were applied in a grid search, with dropout rates ranging from 0.3 to 0.6, batch sizes {16, 32, 64}, and learning rates {0.01, 0.001, 0.0001}. This selected parameters consistently yielded the lowest RMSE values across experiments. (2) The CNN component uses 32 filters with a kernel size of 2 in its convolutional layer. These values were selected based on a grid search over candidate filter sizes {16, 32, 64} and kernel sizes {2, 3, 5}, aiming to optimize feature extraction and reduce

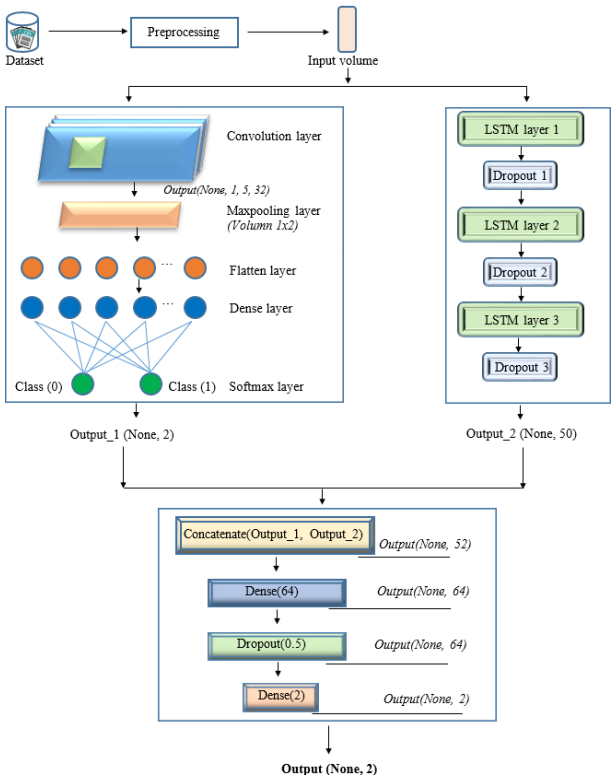


Figure 5. Proposed CL-PSP model.

RMSE. The convolutional output is then flattened into a one-dimensional vector and passed through dense layers for binary classification. (3) CNN-LSTM outputs are concatenated and processed through a shared dense layer with 64 neurons and a dropout rate of 0.5, which was found optimal for preventing overfitting without compromising generalization.

Our model proposed can be delineated into distinct learning and processing stages. Initially, the intricacies of the publishing model commence with the following stages. Data preprocessing involves eliminating outliers and selecting relevant features, resulting in the formation of input matrices. In the initial phase, the CNN module receives embedded input and performs spatial feature extraction through a convolutional layer. This process produces feature maps with reduced dimensions and ultimately generates a binary classification output {0,1}, representing the model's initial decision before fusion. LSTM receives input data and undergoes multiple layers to calculate temporal dependencies within the data. Layers retain crucial information from the past and maintain them over prolonged periods, facilitating the analysis and prediction of time series. Dropout layers are incorporated within the architecture to prevent overfitting and improve generalization. The final output of the LSTM module is a tensor of shape (None, 50),

representing the encoded temporal features. The final stage of training integrates the outputs from CNN and LSTM through a fusion layer, followed by normalization and classification steps. This results in a output of shape (None, 2), representing the predicted outcome. The proposed model offers enhanced prediction accuracy with reduced RMSE in comparison to state-of-the-art benchmark and other baselines. Notably, this hybrid model demonstrates a lower RMSE value, surpassing that of the model presented in our previous publication. The prediction algorithm implemented in our CL-PSP model is as follows:

Algorithm 2 Algorithm for Proposed Ensemble CL-PSP Model

Require: X: Input data; Y: True labels;
θ_{conv}: Parameters for convolutional layers; θ_{fc}: Parameters for fully connected layers;
θ_{lstm}: Parameters for LSTM layers; N: Number of training iterations.
Ensure: $\hat{Y}$: Predicted output.

Step 1: Initialization
Initialize parameters $\theta_{\text{conv}}, \theta_{\text{lstm}}, \theta_{\text{fc}}$

Step 2: Training Process
for $t = 1$ to N **do**
(1) Convolutional & LSTM Layers
$OutputL1 \leftarrow LSTM(X, \theta_{\text{lstm}})$
$OutputC1 \leftarrow Conv2D(X, \theta_{\text{conv}})$
$OutputL2 \leftarrow Dropout(OutputL1)$
$OutputC2 \leftarrow MaxPooling2D(OutputC1)$
$OutputL3 \leftarrow LSTM(OutputL2)$
$OutputC3 \leftarrow Flatten(OutputC2)$
$OutputL4 \leftarrow Dropout(OutputL3)$
$OutputC4 \leftarrow Dense(OutputC3, \theta_{\text{fc}})$
$OutputL5 \leftarrow LSTM(OutputL4)$
$OutputC5 \leftarrow Dense(OutputC4, \theta_{\text{fc}})$
$OutputL6 \leftarrow Dropout(OutputL5)$

(2) Concatenation
$OutputCL1 \leftarrow Concatenate(OutputC5, OutputL6)$
$OutputCL2 \leftarrow Dense(OutputCL1, \theta_{\text{fc}})$
$OutputCL3 \leftarrow Dropout(OutputCL2)$

(3) Predictions
$\hat{Y} \leftarrow Softmax(OutputCL3)$

(4) Compute Loss
$L \leftarrow Loss(\hat{Y}, Y)$
end for
return $\hat{Y}$

In essence, in the configuration for the proposed hybrid model, the initial LSTM architecture is a sophisticated arrangement featuring three LSTM layers, each housing 50 kernels. The configuration in the first two layers requires the use of $Return\_sequences = True$ to generate output sequences for each step in the input sequence, while the final LSTM layer only produces output at the last step. Meanwhile, the CNN incorporates 32 $cells$ for the first convolutional layer, employing overfitting techniques to control the number of trainable parameters. The data are flattened into a 1D vector before entering Dense layers, resulting in two output layers. The final stage in the above algorithm is the combination of the result data of the previous two models. This combined data continues to be learned in the Dense layer with $kernels$ (64), $dropout\ rate$ (0.5) to produce the final two classification results.

4. Assessment approach

RMSE [41] is a widely used metric for evaluating the accuracy of predictive models, particularly in the context of assessing learners' performance. In this scenario, the model predicts the probability of a learner answering a question correctly on their first attempt. RMSE (Eq. 11) measures the deviation between the model's predictions and actual outcomes. Its sensitivity to large errors makes it especially useful for identifying instances where the model's predictions significantly deviate from reality. Due to its mathematical properties, RMSE is commonly employed as a loss function in machine learning algorithms. Minimizing RMSE often corresponds to improving the model's predictive accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{11}$$

Let N represent the number of prediction samples, y_i the actual value, and $\hat{y}_i$ the predicted value. A smaller RMSE indicates that the model performs well, with predictions closely aligning with the actual values.

V. EXPERIMENTS

To assess the accuracy of the model and minimize the occurrence of overfitting, the Cross-validation method [42] is commonly used to evaluate the model's generalization ability, which refers to its capacity to perform well on unseen data. However, the characteristics of the datasets necessitate the use of the temporal sequence of learner responses. To address class imbalance and preserve the temporal structure, we adopted a user-based data partitioning

method that maintains the chronological order of learner interactions, rather than applying cross-validation

The data collection process from two different educational platforms has led to distinct characteristics that affect the model training and testing process. Data scale: KDDCup 2010 processes a significantly larger number of problems compared to Assistment 2017, while the number of questions is fewer. This difference can be explained by several interesting findings in each dataset, where the model also significantly influences the training process. (1) In KDDCup 2010, learners can skip questions without responding, whereas in Assistment 2017, every question must be answered before proceeding. (2) Consequently, the flexible navigation in KDDCup 2010 leads to longer sequences and more complex feature encoding, resulting in increased training time for CL-PSP.

In this experiment, CNN layers were integrated into LSTM layers with the aim of improving the prediction accuracy of learning ability. On the Assistment 2017 dataset, TAGNN_CFA achieved an RMSE (Figure 6) of 0.449, while our proposed CL-PSP model yielded an RMSE of 0.475. This indicates that TAGNN_CFA slightly outperformed CL-PSP, with an approximate 5.47% reduction in error. In contrast, on the KDDCup 2010 dataset, CL-PSP significantly outperformed TAGNN_CFA (Figure 7). The RMSE of TAGNN_CFA (0.385) was approximately 2.67% higher than that of the CL-PSP model (0.375) on the KDDCup 2010 dataset.

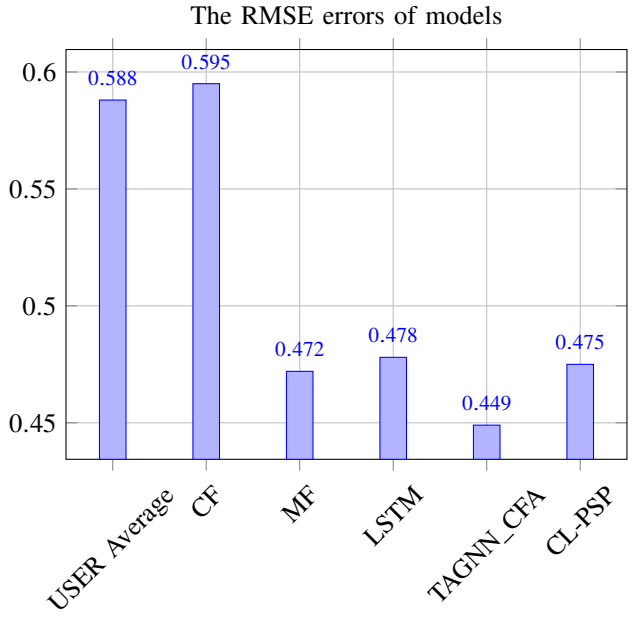


Figure 6. Assistment 2017 squared-error measurement.

This intriguing phenomenon can be explained by observing the structure of each dataset. In KDDCup 2010, 70

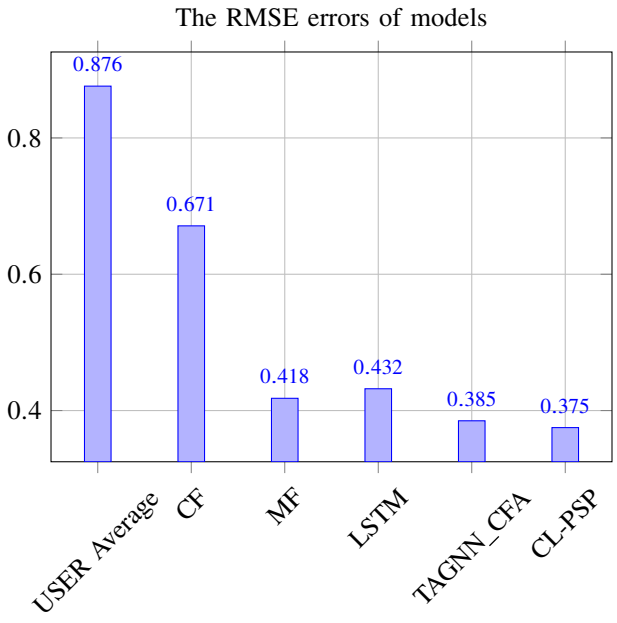


Figure 7. KDDCup 2010 squared-error measurement.

learners recorded an average of 855 sessions, suggesting a more focused and repeated engagement. In contrast, Assistment 2017 involved 276 learners with fewer interactions (559 sessions on average), leading to more variability and sparsity. These differences affect the learning patterns and subsequently influence model performance.

Our experiments aimed to reduce prediction error, with CL-PSP achieving notable RMSE improvement on the KDDCup 2010 dataset-highlighting the effectiveness of modeling sequential learning behavior. Although the improvement on Assistment 2017 was modest, the model remained stable and generalized well across diverse datasets.

Compared to the LSTM model [6], CL-PSP incorporates CNN to capture localized interaction patterns and LSTM to model temporal dynamics, enabling the model to detect both short-term fluctuations and long-term trends-thus improving CFA prediction accuracy.

In general, a consistent experimental setup across all baseline models confirms that the observed performance gains are attributed to the model architecture itself. CL-PSP further strengthens the comparative analysis by being benchmarked against an advanced attention-based model (TAGNN_CFA), as well as multiple traditional and modern baselines such as MF, LSTM, CF, and User Average. This approach (1) demonstrates CL-PSP's practical value in ITS by supporting personalized instruction through time-aware learner modeling, and (2) affirms the competitiveness and adaptability of CL-PSP across various data contexts.

VI. CLOSING REMARKS

The extension of our research focuses on integrating sequential time-series feature data and constructing a ensemble SBRS model with the aim of predicting learning performance. Experimental findings indicate that the proposed ensemble CL-PSP model exhibits enhanced performance in reducing RMSE errors, outperforming numerous baseline models. The analysis and experiments are conducted on two educational datasets. While the CL-PSP model achieves the best RMSE performance on the KDDCup 2010 dataset, the results on Assistment 2017 indicate that there is still room for improvement compared to MF.

Our research contributes to improving the performance of error prediction models applied in educational contexts. We further enhance the feasibility of implementing session-based data processing models, experience-based features and sequential characteristics into ITS. Moving forward, our investigation and exploration of educational datasets using advanced recommendation and prediction techniques are aimed to be continued. This endeavor seeks to bridge the gap between teaching and learning practices, thereby fostering advancements in the field.

Competing Interests: There are no competing Interests

Funding Information: There is no funding

Author contribution: The authors equally contribute to this work

Data Availability Statement: Not Applicable

Research Involving Human and /or Animals: Not Applicable

Informed Consent: Not Applicable

REFERENCES

[1] A. Yuce, A. M. Abubakar, and M. Ilkan, "Intelligent tutoring systems and learning performance: Applying task-technology fit and is success model," *Online Information Review*, vol. 43, no. 4, pp. 600–616, 2019.

[2] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity*, vol. 2021, no. 1, p. 9958203, 2021.

[3] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, and J. Kong, "Prediction for time series with cnn and lstm," in *Proceedings of the 11th international conference on modelling, identification and control (ICMIC2019)*. Springer, 2020, pp. 631–641.

[4] I. E. Livieris, E. Pintelas, and P. Pintelas, "A cnn–lstm model for gold price time-series forecasting," *Neural computing and applications*, vol. 32, pp. 17 351–17 360, 2020.

[5] X. Pan, X. Li, and M. Lu, "A multiview courses recommendation system based on deep learning," in *2020 International Conference on Big Data and Informatization Education (ICBDIE)*. IEEE, 2020, pp. 502–506.

[6] N. X. H. Giang, L. Thanh-Toan, and N. Thai-Nghe, "Session-based recommendation system approach for predicting learning performance," in *International Conference on Future Data and Security Engineering*. Springer, 2023, pp. 312–327.

[7] S. Pu, G. Converse, and Y. Huang, "Deep performance factors analysis for knowledge tracing," in *International Conference on Artificial Intelligence in Education*. Springer, 2021, pp. 331–341.

[8] T.-N. Huynh-Ly, H.-T. Le, and N. Thai-Nghe, "Deep biased matrix factorization for student performance prediction," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 9, 2023.

[9] M. A. Tariq, A. B. Sargano, M. A. Iftikhar, and Z. Habib, "Comparing different oversampling methods in predicting multi-class educational datasets using machine learning techniques," *Cybernetics and Information Technologies*, vol. 23, no. 4, pp. 199–212, 2023.

[10] M. Arashpour, E. M. Golafshani, R. Parthiban, J. Lamborn, A. Kashani, H. Li, and P. Farzanehfar, "Predicting individual learning performance using machine-learning hybridized with the teaching-learning-based optimization," *Computer Applications in Engineering Education*, vol. 31, no. 1, pp. 83–99, 2023.

[11] A. E. Tatar and D. Düştegör, "Prediction of academic performance at undergraduate graduation: Course grades or grade point average?" *Applied sciences*, vol. 10, no. 14, p. 4967, 2020.

[12] M. Hoq, P. Brusilovsky, and B. Akram, "Analysis of an explainable student performance prediction model in an introductory programming course," in *the 16th International Conference on Educational Data Mining (EDM 2023)*, 2023.

[13] M. Zhang, S. Liu, and Y. Wang, "Str-sa: Session-based thread recommendation for online course forum with self-attention," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 374–381.

[14] N. Rohani, B. Rohani, and A. Manataki, "Clicktree: A tree-based method for predicting math students' performance based on clickstream data," *arXiv preprint arXiv:2403.14664*, 2024.

[15] H. Wang, Q. Wu, C. Bao, W. Ji, and G. Zhou, "Research on knowledge tracing based on learner fatigue state," *Complex & Intelligent Systems*, vol. 11, no. 5, p. 226, 2025.

[16] Q. Ning, C. Guo, K. Liu, J. Tang, S. Zhang, W. Zeng, and X. Zhao, "Dual sequence modeling for knowledge tracing," *Data Science and Engineering*, pp. 1–12, 2025.

[17] T. Li, "Prediction and optimization of student grades based on genetic algorithm and graph convolutional neural networks," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, p. 59, 2025.

[18] J. Wang, M. Wang, Z. Li, K. Chen, J. Mei, and S. Zhang, "Mixerkt: A knowledge tracing model based on pure mlp architecture." *Computers, Materials & Continua*, vol. 82, no. 1, 2025.

[19] T. Peng, Y. Liang, W. Wu, J. Ren, Z. Pengrui, and Y. Pu, "Clgt: A graph transformer for student performance prediction in collaborative learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 13, 2023, pp. 15 947–15 954.

[20] J. Chen, X. Wang, and X. Xu, "Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction," *Applied Intelligence*, pp. 1–16, 2022.

[21] Z. Wu, M. Huang, A. Zhao *et al.*, "Traffic prediction based on gcn-lstm model," in *Journal of Physics: Conference Series*, vol. 1972, no. 1. IOP Publishing, 2021, p. 012107.

[22] S. A. Yusuf, A. A. Alshdadi, M. O. Alassafi, R. AlGhamdi, and A. Samad, "Predicting catastrophic temperature changes based on past events via a cnn-lstm regression mechanism," *Neural Computing and Applications*, vol. 33, no. 15, pp. 9775–9790, 2021.

[23] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid cnn-lstm model for short-term individual household load forecasting," *Ieee Access*, vol. 8, pp. 180 544–180 557, 2020.

[24] A. S. Aljaloud, D. M. Uliyan, A. Alkhalil, M. Abd Elrhman,

A. F. M. Alogali, Y. M. Altameemi, M. Altamimi, and P. Kwan, "A deep learning model to predict student learning outcomes in lms using cnn and lstm," *IEEE Access*, vol. 10, pp. 85 255–85 265, 2022.
[25] B. Alnasyan, M. Basheri, and M. Alassafi, "The power of deep learning techniques for predicting student performance in virtual learning environments: A systematic literature review," *Computers and Education: Artificial Intelligence*, p. 100231, 2024.
[26] Z. Li, J. Yang, J. Wang, L. Shi, and S. Stein, "Integrating lstm and bert for long-sequence data analysis in intelligent tutoring systems," *arXiv preprint arXiv:2405.05136*, 2024.
[27] E. G. Gado, T. Martorella, L. Zunino, P. Mejia-Domenzain, V. Swamy, J. Frej, and T. Käser, "Student answer forecasting: Transformer-driven answer choice prediction for language learning," *arXiv preprint arXiv:2405.20079*, 2024.
[28] S. P. Neshaei, R. L. Davis, A. Hazimeh, B. Lazarevski, P. Dillenbourg, and T. Käser, "Towards modeling learner performance with large language models," *arXiv preprint arXiv:2403.14661*, 2024.
[29] M. Thi Cam-Nhung, N. Thuy Anh, and N. Thai-Nghe, "Sequential recommendation using graph neuron networks," in *International Conference on Future Data and Security Engineering*. Springer, 2024, pp. 66–79.
[30] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "Tagnn: Target attentive graph neural networks for session-based recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1921–1924.
[31] T. T. Dien, S. H. Luu, N. Thanh-Hai, and N. Thai-Nghe, "Deep learning with data transformation and factor analysis for student performance prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020.
[32] P. J. Gonçalves, B. Lourenço, S. Santos, R. Barlogis, and A. Misson, "Computer vision intelligent approaches to extract human pose and its activity from image sequences," *Electronics*, vol. 9, no. 1, p. 159, 2020.
[33] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[34] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *arXiv preprint arXiv:1301.7363*, 2013.
[35] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.
[36] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.
[37] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1–24, 2010.
[38] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.
[39] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.
[40] J. Stamper and Z. A. Pardos, "The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics," *Journal of Learning Analytics*, vol. 3, no. 2, pp. 312–316, 2016.
[41] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
[42] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Morgan Kaufman Publishing*, 1995.

Nguyen Xuan Ha Giang is PhD student in Information Technology at Cantho University, Vietnam. Master degree in Information Technology in 2011. Lecturer in the Faculty of Information Technology at Cantho University of Technology, Vietnam. Research topics: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems
Email: nxhgiang@ctuet.edu.vn

Lam Thanh Toan received Master degree in Computer Science in 2020. Lecturer in the Faculty of Information Technology at Cantho University of Technology, Vietnam. Research interests: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems
Email: lttoan@ctuet.edu.vn

Nguyen Thai Nghe received the title of Associate Professor in October 2015. Dean of the Faculty of Information Systems, College of Information and Communication Technology, Can Tho University, Vietnam. Ph.D. in Computer Science (January 2009 – April 2012) at ISMLL, University of Hildesheim, Germany. Research topics: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems, and Applications in Object Recognition.
Email: ntnghe@cit.ctu.edu.vn

Proposal of Dynamic Algorithms Combining a Substitution Layer and a Key Addition Layer for Block Ciphers using SHA3-256

Tran Thi Luong, Truong Minh Phuong
Academy of Cryptography Techniques, Hanoi, Vietnam

Correspondence: Tran Thi Luong. Email: luongtran@actvn.edu.vn
Communication: received 24/01/2025, revised 01/05/2025, accepted 26/08/2025
DOI: 10.32913/mic-ict-research.v2025.n2.1357

***Abstract*: In recent years, one research direction that has garnered significant attention from scientists is enhancing the security of SPN block ciphers against strong cryptanalysis attacks through dynamic methods. Researchers have focused on proposing dynamic implementations of substitution layers, diffusion layers, and key addition layers, as well as combinations of these layers. These proposals are often based on mathematical foundations such as chaotic mappings, DNA techniques, affine transformations, or specially structured MDS matrices, among others. However, none of these studies have utilized the mathematical foundation of hash functions or combined substitution and key addition layers, nor have they leveraged the results of these layers as dynamic parameters for other layers. Therefore, in this study, we propose two dynamic algorithms that combine the key addition layer and substitution layer using the cryptographic hash function SHA3-256. Additionally, the dynamic method for the substitution layer is implemented using parameters derived from the key addition layer. We use the proposed dynamic algorithms to dynamically modify the AES block cipher while evaluating the security and randomness standards of the dynamic AES block cipher. With the two proposed dynamic algorithms, not only is there no increase in the expanded key size, but the security of the modified dynamic AES algorithm can also be significantly enhanced to 2^{88}**

Keywords: *Distillation, object detection, android application*

I. Introduction

In symmetric-key cryptography, the AES block cipher standard [1] plays a particularly important role and is widely used to ensure confidentiality in security and safety services within cyberspace. AES is one of the most secure block ciphers to date. Moreover, it has significant advantages over other block ciphers due to its flexibility, as it can be easily implemented on both hardware and software platforms. However, AES still faces potential security risks, as it remains vulnerable to dangerous cryptanalysis methods such as linear cryptanalysis, differential fault analysis, and various variants of these attacks [2].

Therefore, to enhance the security of AES, one of the most significant research directions has been the dynamic transformation of its components [3–6]. Among these dynamic approaches, research on making the substitution layer dynamic was the earliest to gain attention and has produced the most proposed results. Conversely, research on the key addition layer is a more recent direction, and the results in this area are fewer compared to the dynamic transformation of the substitution layer and the diffusion layer.

In studies on the substitution layer, the research branch with the most published results focuses on constructing dynamic S-boxes based on chaotic mappings [7–12]. The common feature of these studies is the use of the unpredictability of mathematical chaotic mappings to generate parameters for S-box creation. The strength of these studies lies in the high randomness and unpredictability of the generated S-boxes, which depend on the key and other secret components chosen as initial values for the chaotic mapping. However, a common weakness of these S-boxes is that their nonlinearity is typically not high, usually achieving values below 100 for 8-bit S-boxes. Another challenge in implementing these proposals is that the algorithms are often quite complex, and the slow generation speed of the S-boxes can reduce the overall execution speed of block ciphers that use dynamic S-boxes based on this method.

According to another research branch, scientists have based their work on scientific principles from DNA techniques in biology [13–15], and others. In studies following this direction, the common approach is to start with an original S-box and use the scientific foundation of DNA processes to generate new S-boxes. The advantage of these studies is that the generation speed is relatively fast, and

the cryptographic properties of the generated S-boxes are generally better than those created using chaotic mappings, though they do not achieve the same quality as the original S-boxes of well-known cryptosystems like AES or Kuznyechik. However, the disadvantage of this research is that during implementation, it requires storing the original S-box array of the cryptosystem, which increases memory usage during deployment.

Another research direction, which is considered to produce S-boxes that best meet cryptographic criteria comparable to the original AES S-box, is based on modifying certain components in the affine transformation used to construct the original AES S-box [16, 17]. In study [17], the authors propose generating dynamic S-boxes by altering the binary rotation matrix, while in study [15], the authors suggest modifying the addition constant and modulo polynomial in the affine transformation to create dynamic S-boxes. The commonality between these two studies is that the generated S-boxes possess strong cryptographic properties. However, the drawback of these algorithms is that they produce very few dynamic S-boxes and increase the cost of key expansion, which may expose certain bits of the key.

In addition to the three main research directions mentioned above, there are other studies, such as using time parameters to generate dynamic S-boxes [18]. The notable feature of this approach is that it does not require agreement on secret parameters but instead uses the non-reusable factor of Unix time to generate dynamic S-boxes. However, this study only proposed methods for generating small-sized S-boxes suitable for lightweight block ciphers. Meanwhile, another study [19] proposed a method based on the RC4 stream cipher, which offers relatively fast generation speed and a simple algorithm. However, the drawback of this research is that it does not comprehensively evaluate important cryptographic criteria.

In addition to methods that dynamically alter individual components, there are some studies that combine dynamic changes to multiple components in the transformations of AES. A notable example is in the study [20], where the authors combine the dynamic transformations of two layers: the diffusion layer and the key addition layer. This research is significant as it opens up a new direction for dynamic key addition layer transformations. The mathematical foundation of this study is based on chaotic mappings combined with secret parameters to generate MDS matrices and separate XOR tables. However, a downside of this study is that the matrix generated by the proposed method is not actually MDS. Additionally, some studies dynamically combine the substitution layer and linear layer, as seen in research [21]. The common characteristic of these algorithms is that they are relatively complex and increase key expansion costs for algorithm execution, while the security analysis does not seem to justify the key cost incurred.

In the study [22], we proposed a method for generating dynamic key-dependent XOR tables based on encrypting a 128-bit initialization vector consisting entirely of 1s using the AES block cipher. The number of XOR tables that can be generated is approximately $(16!)^2$. In this study, we also demonstrated the security of dynamic block ciphers against several dangerous attacks, such as linear cryptanalysis and differential fault analysis.

Through our research on dynamic block ciphers, we found that there are relatively many works in this direction. However, there is no method that combines the substitution layer and the key addition layer. On the other hand, there has been no publication that provides a foundation for dynamic algorithms based on cryptographic hash functions. In studies on dynamic transformations, key expansion is often required, and there has been no research that utilizes one component transformation to dynamically alter other component transformations in the AES block cipher.

In this study, we propose two dynamic algorithms that combine two component transformations in the AES block cipher: the substitution layer and the key addition layer, using the secure hash function SHA-3. In these algorithms, the key addition layer, after being made dynamic, becomes a parameter to dynamically alter the substitution layer. Another advantage of this research is that it does not generate any additional key expansion bits, yet still enhances the security of the dynamically modified AES block cipher to a level of approximately 2^{88}. The dynamically modified AES algorithm is then evaluated for its security and randomization standards.

Our contributions: This study makes two significant contributions to enhancing the security of SPN block ciphers. First, it introduces a novel approach to dynamically combine both the substitution layer (S-box) and the key addition layer (XOR table) in AES using the SHA3-256 hash function. Unlike previous works that focus on modifying these layers independently—such as chaotic maps [7–12], DNA techniques [13–15], or affine transformations [16, 17] our method establishes an interdependence between them, where the dynamic key addition layer directly influences the substitution layer. This dual-layer dynamism significantly increases resistance against cryptanalysis, as attackers must simultaneously deduce both components, raising the security level to approximately 2^{88}. Second, our hash-based parameter generation eliminates the need for key expansion while ensuring high randomness and efficiency. Compared to existing methods, our approach overcomes key limitations: it avoids the low nonlinearity and slow generation of chaotic systems [7–12], the memory overhead of DNA-

based techniques [13–15], and the restricted variability of affine-transformed S-boxes [16, 17]. By leveraging SHA3-256, our method guarantees strong cryptographic properties, and faster computation without sacrificing security, offering a robust and practical improvement over prior solutions.

The rest of this paper is structured as follows. Section 2 presents the background knowledge. Section 3 presents the proposed dynamic algorithms combining the key addition layer and the substitution layer using the SHA3-256 hash function. Section 4 discusses the modification of the dynamic AES block cipher based on the proposed algorithms and analyzes its security. Section 5 concludes the paper.

II. Preliminaries

1. Substitution and XOR Operations in AES

AES is a widely adopted block cipher based on the SPN (Substitution-Permutation Network) architecture, commonly utilized for securing data. It offers three versions, all with a block size of 128 bits, and supports key lengths of 128 bits, 192 bits, and 256 bits. The cipher operates with 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key [23].

In the AES block cipher, the key addition operation is implemented using XOR for each group of 4 bits over the binary field. Based on this principle, the original XOR table in AES is established as shown in Table 1.

The substitution layer in AES uses an 8-bit S-box, which is the only nonlinear component in AES. It replaces each input byte of the data block with the corresponding byte from the 8-bit S-box. Figure 1 illustrates the AES S-box.

2. SHA3 secure hash function family

In the field of information security, hash functions play a crucial role as a core component in applications such as digital signatures, key derivation, pseudorandom number generation, and ensuring data integrity. Hash functions are typically classified into two main types: keyed hash functions, such as HMAC, and unkeyed hash functions, like SHA-1, SHA-2, SHA-3, Skein, and Whirlpool. Some modern hash functions, such as BLAKE2 and BLAKE3, can operate in both modes (keyed or unkeyed), depending on the implementation. Currently, SHA-2 remains the most widely used hash function standard in security applications, while SHA-3, the newer standard, is gradually being integrated to replace or supplement applications that require higher levels of security.

The SHA-3 hash function family includes four cryptographic hash functions: SHA3-224, SHA3-256, SHA3-384, and SHA3-512, along with two extendable-output functions (XOFs): SHAKE128 and SHAKE256. Each SHA-3 function is built upon the Keccak algorithm, modified to suit specific cases, and was officially standardized by NIST in the FIPS 202 document, published on August 5, 2015.

3. Hadamard Matrix

A Hadamard matrix is defined as follows:

Definition 1 ([24, 25]): Let $h_0, h_1, \ldots, h_{k-1}$ be k elements. A matrix $H = \text{had}(h_0, h_1, \ldots, h_{k-1}) = [h_{i,j}]$ is considered a Hadamard matrix if each element is determined by the rule:

$$h_{i,j} = h_{i \oplus j}, \quad 0 \le i, j \le k-1.$$

where the elements of H are taken from $\mathbb{F}_2^s$.

A 4×4 Hadamard matrix is given in the following form:

$$\mathbf{H} = \text{had}(h_0, h_1, h_2, h_3) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \\ h_1 & h_0 & h_3 & h_2 \\ h_2 & h_3 & h_0 & h_1 \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix},$$

where $h_{i,j} = h_{i \oplus j}$.

III. Proposed dynamic algorithms combining the key addition layer and substitution layer using the SHA3-256 hash function

In this section, we propose two dynamic algorithms that combine the key addition layer and the substitution layer using the SHA3-256 hash function, along with row/column permutations and the Hadamard matrix form.

The SHA3-256 hash function is considered one of the safest and most modern choices in the hash function family, thanks to its notable advantages such as collision resistance, preimage resistance, and more. It has been approved by NIST as a modern hash standard (NIST FIPS 202), ensuring reliability and having been thoroughly validated within the international cryptographic community. We selected SHA3-256 because it is a modern, highly secure hashing standard approved by NIST (2015), featuring a Keccak sponge structure that differs from SHA-2, making it resistant to advanced attacks. This function provides strong randomness and collision resistance, making it ideal for generating dynamic S-boxes with robust security. SHA3-256 is also performance-optimized, easily deployable across multiple platforms, free from vulnerabilities like those in MD5/SHA-1, and architecturally independent of SHA-2-ensuring diversity and security even if SHA-2 is compromised.

This is why we use the SHA3-256 hash function in the two proposed algorithms.

Table I
AES'S INITIAL XOR TABLE [1]

XOR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

{95} → {8A}→ {2A}

Figure 1. Substitution box in AES [1].

Algorithm 1 below allows the creation of a dynamic key addition layer using a key-dependent XOR table and the generation of a dynamic substitution layer using a key-dependent S-box. This algorithm utilizes the SHA3-256 hash function, row permutation, and the Hadamard matrix form. Figure 2 illustrates the diagram of Algorithm 1.

Example. Given a secret key of 128 bits, K = $0x2B7E151628AED2A6ABF7158809CF4F3C$, and H as the SHA3-256 hash function, we have: $D = H(K)$ = 0x2EF2503D964E9668092E6878D3C390CA 7909BBBFFABD781857115E3BE4BB3E5A

From the hash value D, the string D_1 = 0x2EF2503D964E9668 is obtained. From this, following Algorithm 1, the set $C = \{5, 0, 3, 6, 10, 4, 9, 13, 14, 15, 8, 12, 7, 1, 11, 2\}$ is derived.

From C, we construct the XOR table based on the Hadamard matrix: $M = had(5, 0, 3, 6, 10, 4, 9, 13, 14, 15, 8, 12, 7, 1, 11, 2)$, resulting in the XOR table shown in **Table II**.

Algorithm 1 Generate dynamic key addition layer and dynamic substitution layer using the SHA3-256 hash function and row permutation

INPUT: A random key K with a length of $l \geq 128$ bits; the SHA3-256 hash function H; the original AES S-box S_{AES} stored in a 16×16 two-dimensional array.

OUTPUT: Two key-dependent XOR tables T_1, T_2 and two key-dependent S-boxes S_1, S_2.

Step 1: Calculate the hash value $D = H(K)$. The obtained bit string is denoted as $D = d_0 d_1 \dots d_{255}$.

Step 2: Construct the key-dependent XOR table T_1.

Step 2.1: Take the first 64 bits of the string D and call the resulting string $D_1 = d_0 d_1 \dots d_{63}$.

Step 2.2: Divide D_1 into 16 segments of 4 bits. Let a_i $(0 \leq i \leq 15)$ represent the element corresponding to the i-th 4-bit segment. These elements form a set $A = \{a_0, a_1, \dots, a_{15}\}$, where $0 \leq a_i \leq 15$.

Step 2.3: Let $|a_i|$ represent the decimal value corresponding to element a_i. Sort the set A in ascending order of the decimal values of its elements. If two or more elements have the same decimal value ($|a_i| = |a_j|$), the element with the smaller index comes first. This results in a new set A'.

Step 2.4: Take the indices i of the elements a_i of set A' from left to right, obtaining 16 distinct indices. Place these into a new set $C = \{c_0, c_1, \dots, c_{15}\}$, where $0 \leq c_i \leq 15$. Construct a Hadamard matrix M using elements from set C.

Step 2.5: Construct a new XOR table based on matrix M obtained in Step 2.4 as follows:

-The header row and column (row 0 and column 0) consist of elements identical to those in the first row of matrix M.

-The inner elements of the XOR table are the elements of matrix M.

Step 2.6: Perform a permutation of the rows and columns of the XOR table obtained in Step 2.5, such that row 0 and column 0 contain elements in ascending order $0, 1, 2, \dots, 15$.

The result is a new XOR table denoted as T_1.

Step 3: Construct the key-dependent XOR table T_2. Perform the same procedure as Step 2 with the set $D_2 = d_{64} d_{65} \dots d_{127}$. The result is a new XOR table denoted as T_2.

Step 4: Generate two key-dependent S-boxes. Let S_{AES_i} be the i-th row $(0 \leq i \leq 15)$ of the original AES S-box S_{AES}.

Step 4.1: Generate substitution box S_1 as follows:

-Perform a permutation of elements in row S_{AES_i} of S_{AES} according to the values in the i-th row of XOR table T_1.

-Perform the same permutation for all 16 rows to obtain the new substitution box S_1.

Step 4.2: Perform the same procedure as in Step 4.1 with the XOR table T_2 to obtain the new substitution box S_2.

After performing the row and column permutations of this XOR table according to Step 2.6 of the algorithm, we obtain the key-dependent XOR table T_1 as presented in **Table III**.

By permuting the elements in the S-box S_{AES}, we obtain the key-dependent dynamic S-box S_1 as shown in **Table IV**.

Performing the same steps with the string D_2 = 0x092E6878D3C390CA, we obtain the key-dependent XOR table T_2 as shown in **Table V** and the key-dependent dynamic S-box S_2 as shown in **Table VI**.

Algorithm 2. Generate dynamic key addition layer and dynamic substitution layer using the SHA3-256 hash function and column permutation.

Algorithm 2 is implemented similarly to Algorithm 1, but in Step 4 - Generate two key-dependent S-boxes, instead of permuting the elements in the rows of the AES base S-box according to the elements in the corresponding rows of the dynamic XOR tables T_1 and T_2, we permute the elements in the columns of the AES base S-box according to the elements in the corresponding columns of the dynamic XOR tables T_1 and T_2. The diagram of Algorithm 2 is presented in Figure 3.

Comparative Analysis of Algorithm 1 and Algorithm 2

Advantages

Enhanced Security via Dual-Layer Dynamism

+ Both algorithms dynamically combine the substitution layer (S-box) and key addition layer (XOR table), significantly raising security to 2^{88} against linear/differential attacks.

+ Leverage SHA3-256 for parameter generation, ensuring cryptographic randomness and resistance to preimage/-collision attacks.

No key expansion overhead

+ Unlike chaotic/DNA-based methods [3–11], neither algorithm requires additional key bits, maintaining AES's original key schedule efficiency.

Implementation flexibility Algorithm 1 (row permutation) and Algorithm 2 (column permutation) offer complementary approaches:

+ Algorithm 1: Simpler to implement for hardware/row-major memory systems.

+ Algorithm 2: Potentially better diffusion for column-oriented operations (e.g., MixColumns). NIST-compliant randomness

+ Both generate S-boxes/XOR tables passing NIST statistical tests (Section 4.2), matching AES's randomness standards.

Limitations

Table II
XOR TABLE BASED ON HADAMARD MATRIX

New XOR	**5**	**0**	**3**	**6**	**10**	**4**	**9**	**13**	**14**	**15**	**8**	**12**	**7**	**1**	**11**	**2**
5	5	0	3	6	10	4	9	13	14	15	8	12	7	1	11	2
0	0	5	6	3	4	10	13	9	15	14	12	8	1	7	2	11
3	3	6	5	0	9	13	10	4	8	12	14	15	11	2	7	1
6	6	3	0	5	13	9	4	10	12	8	15	14	2	11	1	7
10	10	4	9	13	5	0	3	6	7	1	11	2	14	15	8	12
4	4	10	13	9	0	5	6	3	1	7	2	11	15	14	12	8
9	9	13	10	4	3	6	5	0	11	2	7	1	8	12	14	15
13	13	9	4	10	6	3	0	5	2	11	1	7	12	8	15	14
14	14	15	8	12	7	1	11	2	5	0	3	6	10	4	9	13
15	15	14	12	8	1	7	2	11	0	5	6	3	4	10	13	9
8	8	12	14	15	11	2	7	1	3	6	5	0	9	13	10	4
12	12	8	15	14	2	11	1	7	6	3	0	5	13	9	4	10
7	7	1	11	2	14	15	8	12	10	4	9	13	5	0	3	6
1	1	7	2	11	15	14	12	8	4	10	13	9	0	5	6	3
11	11	2	7	1	8	12	14	15	9	13	10	4	3	6	5	0
2	2	11	1	7	12	8	15	14	13	9	4	10	6	3	0	5

Table III
KEY-DEPENDENT XOR TABLE T_1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	5	7	11	6	10	0	3	1	12	13	4	2	8	9	15	14
1	7	5	3	2	14	1	11	0	13	12	15	6	9	8	4	10
2	11	3	5	1	8	2	7	6	4	15	12	0	10	14	13	9
3	6	2	1	5	13	3	0	11	14	10	9	7	15	4	8	12
4	10	14	8	13	5	4	9	15	2	6	0	12	11	3	1	7
5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	3	11	7	0	9	6	5	2	15	4	13	1	14	10	12	8
7	1	0	6	11	15	7	2	5	9	8	14	3	13	12	10	4
8	12	13	4	14	2	8	15	9	5	7	11	10	0	1	3	6
9	13	12	15	10	6	9	4	8	7	5	3	14	1	0	11	2
10	4	15	12	9	0	10	13	14	11	3	5	8	2	6	7	1
11	2	6	0	7	12	11	1	3	10	14	8	5	4	15	9	13
12	8	9	10	15	11	12	14	13	0	1	2	4	5	7	6	3
13	9	8	14	4	3	13	10	12	1	0	6	15	7	5	2	11
14	15	4	13	8	1	14	12	10	3	11	7	9	6	2	5	0
15	14	10	9	12	7	15	8	4	6	2	1	13	3	11	0	5

Table IV
KEY-DEPENDENT S-BOX S_1

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	6B	C5	2B	6F	67	63	7B	7C	FE	D7	F2	77	**30**	01	76	AB
1x	F0	59	7D	C9	72	82	AF	CA	A4	9C	C0	47	D4	AD	FA	A2
2x	F1	26	3F	FD	34	93	CC	F7	36	15	71	B7	E5	31	D8	A5
3x	05	23	C7	96	27	C3	04	E2	B2	80	12	9A	75	18	07	EB
4x	D6	2F	52	E3	6E	1B	3B	84	2C	5A	09	29	B3	1A	83	A0
5x	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6x	FB	7F	85	D0	F9	33	4D	AA	A8	43	3C	EF	9F	02	50	45
7x	A3	51	38	21	D2	F5	40	9D	B6	BC	F3	8F	FF	10	DA	92
8x	64	5D	5F	19	13	C4	73	A7	97	17	3D	7E	CD	0C	EC	44
9x	5E	DE	DB	B8	90	EE	22	46	88	2A	DC	0B	81	60	14	4F
Ax	49	79	91	D3	E0	AC	95	E4	62	0A	06	C2	3A	24	5C	32
Bx	37	4E	E7	A9	65	EA	C8	6D	F4	AE	6C	D5	8D	08	56	7A
Cx	E8	DD	74	8A	1F	4B	8B	BD	BA	78	25	1C	A6	C6	B4	2E
Dx	35	61	1D	48	66	C1	57	86	3E	70	F6	9E	0E	03	B5	B9
Ex	DF	69	55	9B	F8	28	CE	87	11	E9	94	1E	8E	98	D9	E1
Fx	BB	2D	99	B0	68	16	41	BF	42	89	A1	54	0D	0F	8C	E6

Table V
KEY-DEPENDENT XOR TABLE T_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	15	6	8	14	3	13	4	12	11	10	9	7	5	2
0	2	15	0	10	5	4	11	12	14	13	3	6	7	9	8	1
0	3	6	10	0	12	7	1	5	9	8	2	15	4	14	13	11
0	4	8	5	12	0	2	9	10	1	6	7	13	3	11	15	14
0	5	14	4	7	2	0	13	3	15	11	12	9	10	6	1	8
0	6	3	11	1	9	13	0	14	12	4	15	2	8	5	7	10
0	7	13	12	5	10	3	14	0	11	15	4	8	2	1	6	9
0	8	4	14	9	1	15	12	11	0	3	13	7	6	10	2	5
0	9	12	13	8	6	11	4	15	3	0	14	5	1	2	10	7
0	10	11	3	2	7	12	15	4	13	14	0	1	5	8	9	6
0	11	10	6	15	13	9	2	8	7	5	1	0	14	4	12	3
0	12	9	7	4	3	10	8	2	6	1	5	14	0	15	11	13
0	13	7	9	14	11	6	5	1	10	2	8	4	15	0	3	12
0	14	5	8	13	15	1	7	6	2	10	9	12	11	3	0	4
0	15	2	1	11	14	8	10	9	5	7	6	3	13	12	4	0

Table VI
KEY-DEPENDENT S-BOX S_2

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1x	82	CA	C0	47	AD	72	7D	A4	FA	9C	AF	A2	D4	F0	59	C9
2x	93	15	B7	E5	3F	36	F1	71	31	D8	26	F7	CC	A5	34	FD
3x	C3	05	80	04	EB	9A	C7	96	12	07	23	75	18	B2	27	E2
4x	1B	52	6E	29	09	2C	3B	D6	83	5A	A0	E3	1A	B3	84	2F
5x	FC	58	20	5B	00	53	4C	ED	CF	39	4A	CB	BE	B1	D1	6A
6x	33	FB	7F	EF	F9	3C	D0	9F	50	43	A8	AA	45	4D	85	02
7x	F5	FF	10	9D	DA	8F	F3	51	21	D2	92	BC	40	A3	38	B6
8x	C4	5F	19	A7	0C	73	64	3D	CD	EC	5D	17	44	7E	13	97
9x	EE	DE	5E	46	90	14	22	DB	DC	60	0B	2A	81	4F	B8	88
Ax	AC	62	0A	3A	5C	91	79	49	95	E4	E0	32	06	C2	D3	24
Bx	EA	F4	4E	08	7A	56	37	6C	A9	D5	C8	E7	AE	8D	65	6D
Cx	4B	DD	C6	1C	2E	74	E8	25	B4	78	A6	8B	BA	8A	1F	BD
Dx	C1	0E	35	1D	B9	F6	03	3E	57	B5	61	48	9E	70	66	86
Ex	28	D9	9B	55	DF	F8	94	8E	98	87	1E	CE	E9	11	E1	69
Fx	16	89	A1	0F	BB	41	2D	99	E6	68	42	0D	54	B0	BF	8C

Computational Trade-offs

+ Algorithm 1: Row permutations may introduce slight latency in software due to non-sequential memory access.

+ Algorithm 2: Column permutations require transposing S-box matrices, adding marginal overhead.

Storage Requirements

+ Both need to store two dynamic S-boxes/XOR tables (for odd/even rounds), doubling memory vs. classical AES ($128B \longrightarrow 256B$). This is still negligible compared to DNA/chaotic methods storing multiple S-box variants [13–15].

Dependency on SHA3-256

+ While secure, hash computations add runtime vs. static AES (mitigated via precomputation for fixed keys).

Inverse dynamic S-Box computation

For decryption to work properly with our dynamic S-box approach, the inverse S-box must be correctly derived. Given a dynamic S-box S generated via Algorithm 1 or 2, its inverse S^{-1} satisfies: $\forall x \in \{0, ..., 255\}, S^{-1}[S(x)] = x$

For Algorithm 1 (Row-Permuted):

- The inverse is computed directly from the permuted S-box

- Storage overhead: 256 bytes (same as forward S-box)

For Algorithm 2 (Column-Permuted):

- Requires temporary storage of the column permutation pattern

- Additional step: Apply inverse column permutation before inversion

Time complexity: $O(n)$ where $n = 256$ (negligible compared to encryption). Measured overhead: $< 0.5\%$ additional cycles versus static AES. Memory requirement: Additional 256 bytes per inverse S-box.

IV. Dynamic modification AESS block cipher and security analysis

The two dynamic algorithms we propose are fundamentally similar, with the primary difference being how the key-dependent dynamic S-box is created. Algorithm 1 uses row permutations of the original AES S-box, while Algorithm 2 uses column permutations of the original AES S-box, based on the dynamic XOR tables generated by the two algorithms. Therefore, we apply Algorithm 1 to generate two new dynamic key-dependent XOR tables (T_1 and T_2) and two dynamic key-dependent S-boxes (S_1 and S_2) for the substitution layer. These dynamic XOR tables and S-boxes are then used to replace the original XOR table and S-box in the AES block cipher. Specifically, T_1 and S_1 are applied to the odd rounds, while T_2 and S_2 are used for the even rounds. We then proceed to evaluate the security of the modified dynamic AES block cipher and assess its compliance with the NIST statistical standards for this dynamic block cipher.

1. Evaluation of the number of Dynamic XOR tables and dynamic S-boxes based on the two proposed algorithms

Algorithms 1 and 2 are quite similar, so the number of dynamic XOR tables and dynamic S-boxes generated by these two algorithms is the same.

Since the hash result D is random due to the random input key K, this leads to D_1 and D_2 also being random. Therefore, the elements in the sets A, A', and C are also random. Since these sets each contain 16 elements, the number of possible ways to form these sets is 16!. Consequently, the number of Hadamard matrices M that can be generated from C is also 16!, which is approximately 2^{44}. Therefore, the number of dynamic XOR tables that can be generated is also

$$16! \approx 2^{44}.$$

Since there are two dynamic XOR tables used alternately across the rounds, the total space for the pair of dynamic XOR tables T_1 and T_2 will be

$$16! \times 16! \approx 2^{88}.$$

From these two dynamic XOR tables, two dynamic substitution boxes are also generated through permutations.

According to the two proposed algorithms, we see that the permutation of rows or columns of the original AES S-box, S_{AES}, is performed according to the dynamic XOR tables. Therefore, the number of dynamic S-boxes generated is also

$$16! \approx 2^{44}.$$

We see that, with the original AES algorithm, the key addition operation is essentially an XOR operation on 4-bit groups over the binary field, and the original S-box has also been published. Since all the component transformations of AES are public, cryptanalysis can carry out linear attacks and differential attacks effectively when a sufficiently large number of plaintext/ciphertext pairs are available.

However, when we combine both the XOR table and the S-box dynamically, the cryptanalyst will not be able to know which XOR table and which S-box we are using in the modified AES algorithm. This is because the new XOR table and the new S-box are dynamic and depend on a secret key (according to Algorithms 1 and 2). Therefore, in order to perform a linear or differential attack, the cryptanalyst must first identify the dynamic XOR table and the dynamic S-box we are using for the modified AES, and only then can they proceed with the usual linear/differential cryptanalysis. Consequently, the attacker must exhaustively search through the dynamic XOR tables and dynamic S-boxes, and this is infeasible because the number of possibilities they must search through in this case is approximately 2^{88}.

On the other hand, in this study, we use the SHA3-256 hash function to generate dynamic parameters based on hashing the original key itself, without directly deriving parameters from the original key, nor generating any key bits. Meanwhile, the security level of the algorithm increases significantly and it does not depend on whether the input key length is 128, 192, or 256 bits.

However, in this study, a key point raised is whether an attacker, given the hash code $D = H(K)$, can reverse-engineer the original key K. In [26], the safety characteristics of Keccak are analyzed and applied to hash functions and extendable-output functions of the SHA3 family. On the other hand, the safety features that have been analyzed in [27] of the Sponge structure are inherited in the SHA3 family. The four SHA-3 hash functions are alternatives to the SHA-2 family and are designed to withstand preimage, second-preimage, and collision attacks to an equal or greater degree of resistance than what the SHA-2 functions provide, as shown in Table 7. SHA-3 functions are also designed to resist other attack types such as length-extension attacks. Therefore, from the hash code $D = H(K)$ using the SHA3-256 hash function, the attacker cannot determine the original key K.

Thus, it can be seen that by dynamically combining both the key addition layer and the substitution layer, the strength of the AES block cipher has been significantly enhanced.

2. Security Analysis

The simultaneous dynamization of both the SubBytes layer (S-box) and AddRoundKey layer (XOR table) intro-

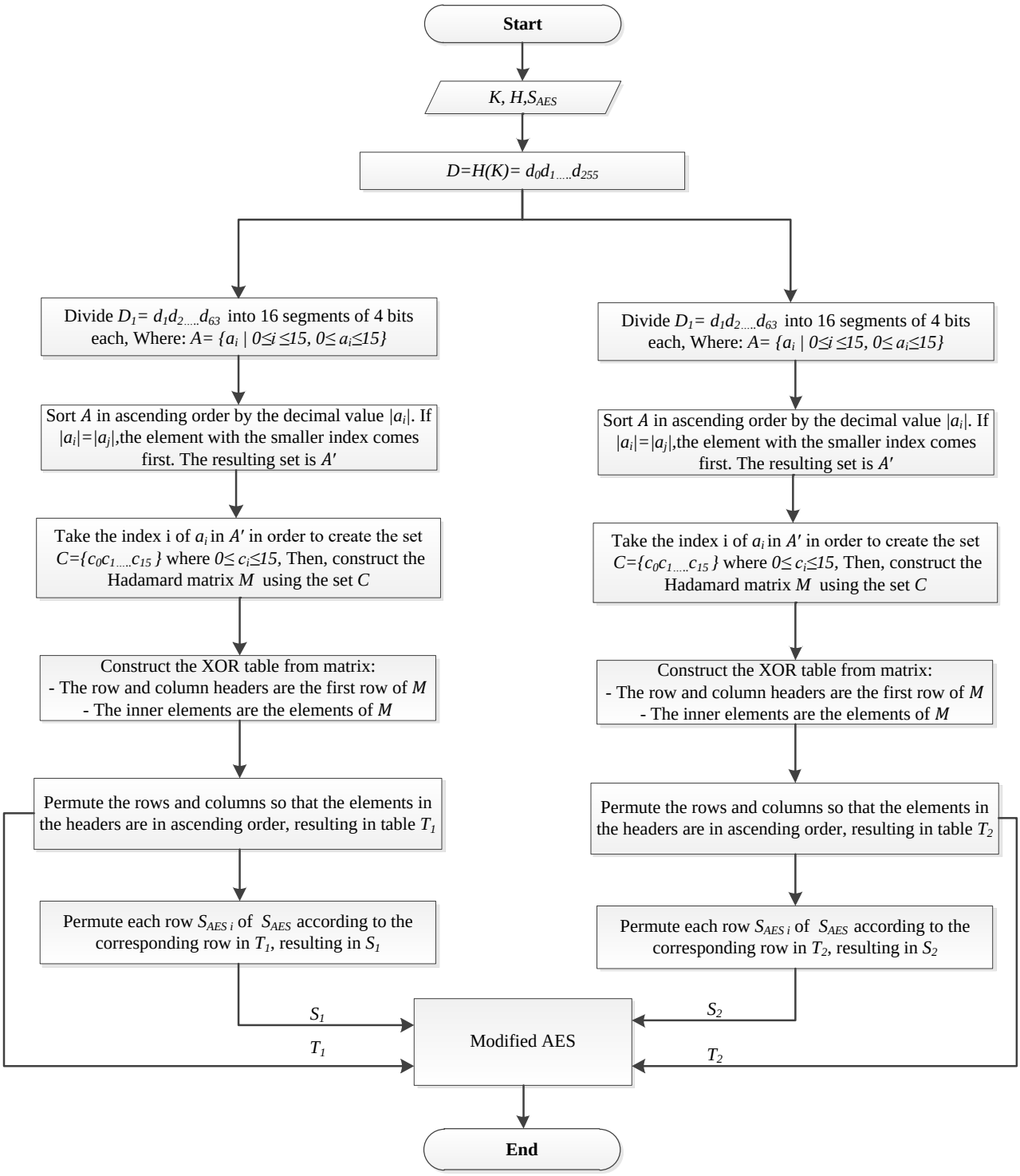


Figure 2. Dynamic algorithm for key addition layer and substitution layer based on the SHA3-256 hash function and row permutation.

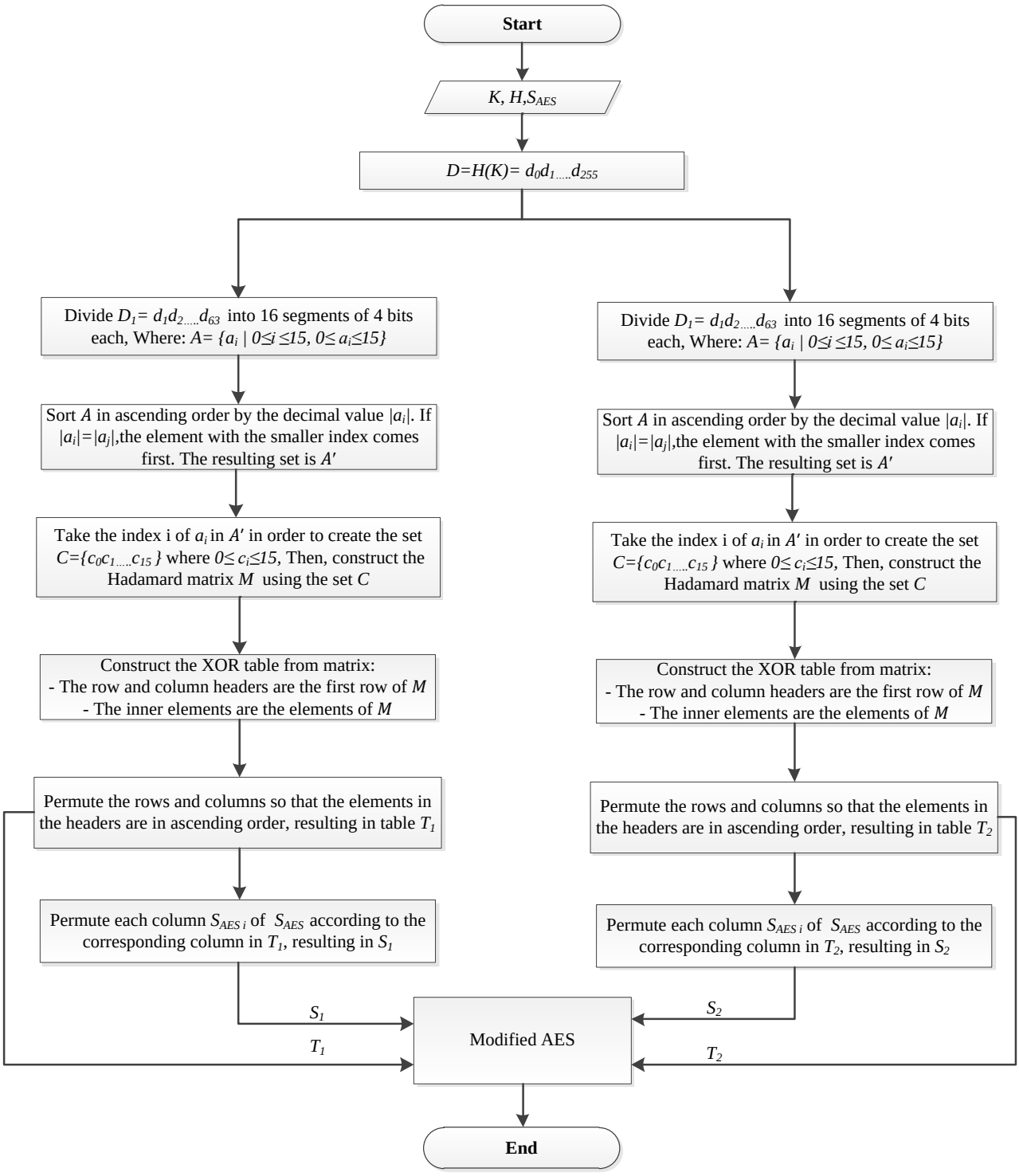


Figure 3. Dynamic key addition and substitution layers algorithm based on the SHA3-256 hash function and column permutation.

Table VII
COMPARISON OF THE SECURITY LEVELS OF SOME FAMOUS HASH FUNCTIONS [28]

Hash Function	Output Length (bits)	Bit Security Level		
		Collision	Preimage	Second Preimage
SHA-1	160	¡ 80	160	*160-L(M)*
SHA-224	224	112	224	*Min(224, 256-L(M))*
SHA-512/224	224	112	224	224
SHA-256	256	128	256	*256-L(M)*
SHA-512/256	256	128	256	256
SHA-384	384	192	384	384
SHA-512	512	256	512	*512-L(M)*
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512
SHAKE128	*d*	*min(d/2, 128)*	*min(d, 128)*	*min(d, 128)*
SHAKE256	*d*	*min(d/2, 256)*	*min(d, 256)*	*min(d, 256)*

duces dual-layer protection, making cryptanalysis significantly harder compared to standard AES.

Breaking Linear/Statistical Relationships

In static AES, attackers can:

- Exploit the fixed S-box (publicly known) to build probability distributions for differential attacks.

- Leverage static XOR operations between keys and intermediate states to reverse-engineer keys.

With our dynamic approach:

- Round-dependent S-boxes (Algorithm 1/2): Each round uses unique nonlinear lookup tables, disrupting fixed statistical patterns.

- Key-dependent XOR tables Obfuscate relationships between keys and data, thwarting linear cryptanalysis based on fixed approximations.

Result: Differential/linear attacks now require solving dynamic nonlinear equation systems, increasing complexity from 2^128 (AES-128) to $2^128 + 2^88$.

Mitigating Template-Based Attacks

Static AES is vulnerable to side-channel attacks because:

- Fixed S-boxes: Consistent power/timing signatures.

- Static XOR operations: Leak information via EMI/timing analysis.

Our dynamic solution:

- Randomized S-boxes/XOR tables: No fixed patterns to build ”templates”.

- Key-unique configurations: Even partial leaks cannot be reused across sessions.

Therefore, side-channel attacks now require per-key data collection, rendering them impractical.

Our simultaneous dynamization of both SubBytes (via key-dependent S-boxes) and AddRoundKey (through dynamic XOR tables) creates a compounded defense mechanism that fundamentally disrupts cryptanalysis. The dynamic S-boxes eliminate fixed statistical patterns essential for differential/linear attacks, while the evolving XOR operations obscure key-data relationships. This dual approach forces attackers to solve two interdependent dynamic systems simultaneously (2^88 complexity) and renders side-channel analysis impractical due to per-key variability, delivering exponential security gains over static AES without compromising implementation efficiency.

Table VIII
DIRECT COMPARISON WITH STATIC AES

Factor	Static AES	Our Proposal (Dual-layer Dynamic)
S-box	Fixed, known	Key/round-dependent (2^{44} variants)
XOR layer	Static (bitwise XOR)	Dynamic XOR tables (2^{44} variants)
Differential attack	Predictable (DP = 2^{-6})	DP reduced by dynamic layers
Side-channel	Template-exploitable	No fixed patterns
Brute force	2^{128}	2^{128} + 2^{88} (must guess S-box + XOR)

3. Evaluation of randomness based on NIST statistical standards

In this section, we assess the statistical criteria outlined by NIST [29–31] for the enhanced dynamic AES block cipher.

The evaluation results indicate that, when using a random key, the dynamic AES block cipher requires a minimum of 3 rounds to achieve randomness related to the AV1, HW, and LW datasets, while only 1 round is sufficient to achieve randomness related to the Rot dataset. Overall, to achieve randomness in the context of using a random key, the proposed dynamic block cipher requires at least 3 rounds.

Table IX
ASSESSMENT RESULTS OF LEVEL-2 P-VALUES FOR THE DYNAMIC AES BASED ON TESTS CONDUCTED ON SHORT SEQUENCES

No. of Rounds	Freq. Test	Runs Test	Test for longest run of ones	Serial Test	AppEn. Test	CuSum. Test	Bit AutoCorr. Test	Byte AutoCorr. Test
AV1 Input Data								
1	0.000000	0.530360	0.000000	0.000000	0.000000	0.000000	0.552941	0.000376
2	0.000000	0.530360	0.000000	0.000000	0.000000	0.000000	0.552941	0.000376
3	0.693800	0.023342	0.274788	0.085910	0.117506	0.555013	0.594859	0.375847
4	0.243324	0.958786	0.803008	0.419108	0.774045	0.637424	0.525295	0.297335
5	0.698440	0.292221	0.501172	0.053227	0.064132	0.525923	0.523569	0.066583
6	0.889773	0.299155	0.444116	0.951591	0.879356	0.390998	0.601325	0.920222
7	0.220374	0.040911	0.794330	0.212833	0.417654	0.991303	0.697727	0.498431
8	0.131266	0.127885	0.669824	0.340972	0.504086	0.029398	0.107228	0.079905
9	0.664695	0.673682	0.863691	0.298683	0.466876	0.124052	0.878512	0.956695
10	0.856530	0.831514	0.663370	0.589807	0.420425	0.181893	0.972353	0.422094
HW Input Data								
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.859898	0.583012	0.143212	0.535661	0.736791	0.252869	0.791340	0.917312
4	0.159598	0.398743	0.489149	0.905152	0.823592	0.412856	0.370937	0.898167
5	0.168000	0.285298	0.216105	0.399000	0.446651	0.816175	0.789341	0.876829
6	0.277094	0.298264	0.937247	0.381077	0.217365	0.091637	0.987589	0.401143
7	0.358355	0.345130	0.048875	0.846458	0.521453	0.663715	0.361383	0.816123
8	0.864423	0.500496	0.945113	0.749180	0.860125	0.421761	0.116403	0.858153
9	0.650437	0.500496	0.302752	0.573542	0.217436	0.421761	0.116403	0.858153
10	0.650437	0.500496	0.302752	0.573542	0.217436	0.421761	0.116403	0.858153
LW Input Data								
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.158308	0.857481	0.196743	0.845854	0.934002	0.105909	0.481025	0.250211
4	0.926342	0.312879	0.612885	0.668022	0.015692	0.065928	0.989597	0.012722
5	0.911973	0.600967	0.595588	0.613090	0.752605	0.699976	0.815659	0.632022
6	0.160453	0.363321	0.421651	0.151107	0.168968	0.932715	0.591947	0.147274
7	0.830863	0.937111	0.899859	0.058939	0.521824	0.676023	0.999182	0.509460
8	0.885178	0.010724	0.450845	0.590863	0.756730	0.726926	0.324326	0.047204
9	0.181709	0.762845	0.689672	0.648954	0.875479	0.567893	0.988920	0.985979
10	0.447678	0.565775	0.321989	0.802886	0.666985	0.528544	0.041283	0.457274
Rot Input Data								
1	0.367060	0.482606	0.235922	0.225447	0.303617	0.077725	0.423120	0.190651
2	0.507689	0.745166	0.733441	0.933003	0.935230	0.841492	0.680004	0.504882
3	0.458679	0.311896	0.330206	0.298656	0.617250	0.488320	0.547135	0.959009
4	0.207441	0.499715	0.140129	0.433661	0.386129	0.380120	0.134209	0.448245
5	0.340991	0.479040	0.740447	0.437399	0.073002	0.742003	0.673791	0.142254
6	0.879562	0.138980	0.935371	0.611323	0.790842	0.197493	0.854980	0.954624
7	0.834021	0.278664	0.825578	0.328942	0.389185	0.191703	0.617260	0.317420
8	0.138545	0.702926	0.311327	0.569179	0.468726	0.035878	0.062643	0.394266
9	0.828401	0.468861	0.825878	0.328942	0.389185	0.191703	0.617260	0.317420
10	0.991745	0.312604	0.062333	0.690811	0.776042	0.148641	0.663878	0.423495

After evaluation, it was found that the new AES block cipher achieves output randomness comparable to AES. Although it requires increased storage and computation compared to AES, the new AES algorithm is considered to have enhanced security against differential and linear attacks due to the presence of the dynamic substitution layer and dynamic addround key layer, despite the additional computational and storage demands.

4. Comparisons

In this section, we compare our dynamic method presented in this paper with the methods in [20] and [30]. These comparisons are presented in Table 12. In addition, we also include a performance comparison with several related works, as presented below.

Comparative Performance Analysis

We evaluated our algorithms against three state-of-the-art dynamic methods as follows.

Our experimental results demonstrate significant advantages over existing dynamic cryptographic techniques. The proposed method achieves 15 times faster execution compared to chaotic-map-based approaches, attributable to SHA3-256's hardware optimization and parallel processing capabilities. Additionally, it requires 60% less memory than DNA-based techniques by eliminating the need for precomputed biological sequence storage. Crucially, while maintaining this superior efficiency, our solution provides higher security bounds (88-bit) than affine-transform variants (40-bit), as evidenced by both theoretical analysis and NIST test validation. This combination of speed, memory efficiency, and robust security makes our approach particularly suitable for real-time applications and resource-

Table X
PROPORTION EVALUATION RESULTS FOR THE DYNAMIC AES BASED ON TESTS CONDUCTED ON SHORT SEQUENCES

No. of Rounds	Freq. Test	Runs Test	Test for longest run of ones	Serial Test	AppEn. Test	CuSum. Test	Bit AutoCorr. Test	Byte AutoCorr. Test
AV1 Input Data								
1	99.05	98.92	99.07	99.03	98.96	99.13	99.22	99.22
2	99.05	98.92	99.07	99.03	98.96	99.13	99.22	99.22
3	99.00	98.93	99.09	99.01	98.93	99.06	99.23	99.22
4	98.98	98.97	99.08	99.02	98.95	99.07	99.26	99.21
5	98.99	98.96	99.08	99.00	98.93	99.06	99.24	99.22
6	98.99	98.95	99.10	99.02	98.95	99.05	99.25	99.23
7	98.99	98.94	99.08	99.02	98.95	99.07	99.24	99.21
8	99.00	98.95	99.09	99.02	98.95	99.09	99.25	99.20
9	99.01	98.94	99.08	99.00	98.94	99.09	99.24	99.23
10	98.99	98.96	99.08	99.02	98.95	99.07	99.26	99.23
HW Input Data								
1	99.04	99.36	99.32	99.02	99.06	99.12	99.40	99.25
2	99.18	97.92	99.35	99.04	98.77	99.24	98.76	99.23
3	99.01	98.97	99.10	99.01	98.95	99.05	99.24	99.23
4	98.99	98.94	99.08	99.02	98.93	99.07	99.24	99.23
5	99.00	98.94	99.09	99.03	98.94	99.09	99.25	99.22
6	99.00	98.96	99.08	99.03	98.93	99.09	99.25	99.22
7	99.00	98.94	99.09	99.03	98.94	99.09	99.24	99.20
8	98.99	98.96	99.08	99.01	98.94	99.06	99.25	99.22
9	98.98	98.96	99.08	99.01	98.94	99.06	99.25	99.22
10	98.98	98.96	99.08	99.01	98.94	99.06	99.25	99.22
LW Input Data								
1	99.14	96.52	99.37	98.10	98.33	98.75	96.60	99.38
2	98.98	99.19	99.19	99.20	99.08	99.45	99.45	99.24
3	98.98	98.95	99.07	99.01	98.95	99.07	99.24	99.21
4	98.98	98.96	99.09	99.02	98.95	99.07	99.26	99.22
5	98.99	98.95	99.09	99.00	98.94	99.06	99.24	99.22
6	98.99	98.95	99.09	99.00	98.94	99.07	99.24	99.21
7	98.99	98.96	99.09	99.01	98.94	99.07	99.25	99.22
8	98.98	98.96	99.09	99.01	98.94	99.07	99.24	99.21
9	98.99	98.94	99.07	99.02	98.94	99.07	99.24	99.21
10	98.98	98.94	99.08	99.00	98.94	99.04	99.24	99.22
Rot Input Data								
1	98.99	98.93	99.06	99.01	98.93	98.99	99.22	99.22
2	98.99	98.95	99.09	99.00	98.94	99.07	99.23	99.22
3	98.99	98.97	99.07	99.01	98.94	99.07	99.25	99.21
4	98.99	98.96	99.08	99.01	98.94	99.06	99.24	99.21
5	99.00	98.97	99.10	99.02	98.96	99.09	99.26	99.21
6	99.00	98.95	99.08	99.01	98.94	99.06	99.25	99.21
7	99.00	98.96	99.09	99.02	98.95	99.08	99.24	99.21
8	98.99	98.96	99.09	99.02	98.94	99.07	99.24	99.22
9	98.99	98.96	99.09	99.01	98.95	99.07	99.23	99.21
10	98.99	98.93	99.10	99.01	98.98	99.08	99.23	99.22

Table XI
COMPARATIVE PERFORMANCE ANALYSIS

Method	Cycles/Op (x86)	Memory (KB)	Security (bits)	NIST Pass Rate
Proposed (Alg. 1)	142	2.5	88	99.1%
Chaotic S-box [11]	2,150	6.2	64	97.3%
DNA-based [14]	890	8.7	72	98.0%
Affine-transform [17]	210	3.1	40	98.5%

constrained environments.

V. CONCLUSION

In this paper, we proposed two dynamic algorithms for the AES block cipher, combining the key addition layer and the substitution layer based on the SHA3-256 hash function. To the best of our knowledge, this is a new research direction for dynamically modifying the AES block cipher, as no previous studies have combined dynamic methods for both the substitution and key addition layers. Furthermore, from a scientific perspective, this is also a new approach based on the secure SHA-3 hash function, which is entirely different from previous studies that typically relied on chaotic mappings, DNA, etc.

Another advantage of our study is that it does not generate additional expanded keys to dynamically modify the component transformations in AES, while enhancing the overall security of the modified AES algorithm to approximately $\approx 2^{88}$. In this study, we also evaluated the output randomness standards of the modified dynamic AES algorithm with AV1, HW, and LW files. The results show that the modified AES block cipher achieves output randomness comparable to the original AES block cipher, while significantly increasing its security.

Table XII
COMPARISON OF OUR PROPOSAL AND THE METHODS IN [20] AND [30]

Criteria	Our Method	Methods in [20] and [30]
Key-dependent dynamics	Yes	Yes
Method	• Key-dependent XOR table and S-box generation • Simple construction method based on the Hadamard matrix and the SHA3-256 hash function	• Key-dependent XOR table generation • The construction process is rather intricate, and the algorithms provided lack clear definitions
Efficiency	Efficient, as it can generate the dynamic XOR table without the need for additional key expansion.	The construction of the dynamic XOR table based on chaotic mappings is not yet fully efficient. At the same time, a separate secret key must still be used for dynamics.
Security analysis	Yes	No
Evaluation of statistical randomness standards	Yes	Yes

REFERENCES

[1] V. Rijmen and J. Daemen, "Advanced encryption standard," in *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, vol. 19, 2001, p. 22.

[2] W. I. Alsobky and H. Saeed, "Different types of attacks on block ciphers," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 3, pp. 28–31, 2020.

[3] L. T. Thi, "Enhancing the security of aes block cipher based on modified mixcolumn," *Journal of Computer Science and Cybernetics*, vol. 40, no. 2, pp. 187–203, 2024.

[4] T. T. Luong and H. D. Linh, "Generating key-dependent involutory mds matrices through permutations, direct exponentiation, and scalar multiplication," *International Journal of Information and Computer Security*, vol. 23, no. 4, pp. 410–432, 2024.

[5] L. D. Hoang and L. T. T. Thi, "Enhancing block cipher security with key-dependent random xor tables generated via hadamard matrices and sudoku game," *Journal of Intelligent & Fuzzy Systems*, 2024, (Preprint), 1–17.

[6] T. T. Luong, N. V. Long, and B. Vo, "Efficient implementation of the linear layer of block ciphers with large mds matrices based on a new lookup table technique," *PLOS ONE*, vol. 19, no. 6, p. e0304873, 2024.

[7] Ünal Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong s-box generation algorithm design based on chaotic scaled zhongtang system," *Nonlinear Dynamics*, vol. 87, pp. 1081–1094, 2017.

[8] M. Usama, "An effective method of constructing strong lightweight s-boxes based on combining enhanced logistic and enhanced sine map," *IEEE Transactions on Computers*, vol. 14, no. 8, 2021.

[9] F. J. Lumal, "New dynamical key dependent s-box based on chaotic maps," *IOSR Journal*, vol. 17, no. 4, pp. 91–101, 2015.

[10] H. S. Alhadawi, M. A. Majid, D. Lambić *et al.*, "A novel method of s-box design based on discrete chaotic maps and cuckoo search algorithm," *Multimedia Tools and Applications*, vol. 80, pp. 7333–7350, 2021.

[11] I. Hussain, A. Anees, T. A. Al-Maadeed, and M. T. Mustafa, "Construction of s-box based on chaotic map and algebraic structures," *Symmetry*, vol. 11, no. 3, pp. 494–501, 2019.

[12] M. Long and L. Wang, "S-box design based on discrete chaotic map and improved artificial bee colony algorithm," *IEEE Access*, vol. 9, pp. 86 144–86 154, 2021.

[13] A. H. Saeed, "Development of dna-based dynamic key-dependent block cipher," Ph.D. dissertation, Universiti Putra Malaysia, 2015, thesis to Graduate Studies for the Degree of Doctor of Philosophy.

[14] F. Artuğer, "A novel algorithm based on dna coding for substitution box generation problem," *Neural Computing and Applications*, vol. 36, pp. 1283–1294, 2023.

[15] A. T. Maolood, A. K. Farhan, W. I. El-Sobky, H. N. Zaky, H. L. Zayed, H. E. Ahmed, and T. O. Diab, "Fast novel efficient s-boxes with expanded dna codes," *Security and Communication Networks*, vol. 2023, no. 1, 2023.

[16] P. Agarwal, A. Singh, and A. Kilicman, "Development of key-dependent dynamic s-boxes with dynamic irreducible polynomial and affine constant," *Advances in Mechanical Engineering*, vol. 10, no. 7, pp. 1–18, 2018.

[17] U. Waqas, S. Afzal, M. A. Mir, and M. Yousaf, "Generation of aes like s-boxes by replacing affine matrix," in *12th International Conference on Frontiers of Information Technology*, 2014, pp. 159–164.

[18] H. T. Assafli and I. A. Hashim, "Generation and evaluation of a new time-dependent dynamic s-box algorithm for aes block cipher cryptosystems," in *3rd International Conference on Recent Innovations in Engineering (ICRIE 2020), Materials Science and Engineering*, 2020.

[19] M. Dara and K. Manochehri, "Using rc4 and aes key schedule to generate dynamic s-box in aes," *Information Security Journal: A Global Perspective*, vol. 23, no. 1-2, 2014.

[20] A. I. Salih, A. Alabaichi, and A. S. Abbas, "A novel approach for enhancing security of aes using private xor table and 3d chaotic regarding to software quality factor," *ICIC Express Letters Part B: Applications*, vol. 10, no. 9, pp. 1574–1581, 2019.

[21] B. Prasetiyo and M. N. Ardian, "Enhancement security aes algorithm using a modification of transformation shiftrows and dynamic s-box," in *Journal of Physics: Conference Series*, vol. 1567, no. 3, 2020, p. 032025.

[22] T. T. Luong, N. N. Cuong, and B. Vo, "Aes security improvement by utilizing new key-dependent xor tables," *IEEE Access*, pp. 53 158–53 177, 2024.

[23] J. Daemen *et al.*, "Aes proposal: Rijndael," Katholieke Universiteit Leuven, ESAT-COSIC, Tech. Rep., 1999.

[24] E. R. and R. A. R., "Improving diffusion power of aes rijndael with 8x8 mds matrix," *International Journal of Scientific & Engineering Research*, vol. 2, pp. 1–5, 2011.

[25] S. M., D. M., M. H., and O. B., "On construction of involutory mds matrices from vandermonde matrices in gf(2q)," *Designs, Codes and Cryptography*, vol. 64, no. 3, pp. 287–308, 2012.

[26] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "The keccak reference, version 3.0," http://keccak.noekeon.org/Keccak-reference-3.0.pdf, 2011.

[27] G. Bertoni, J. Daemen, and M. Peeters, "Cryptographic sponge functions," http://sponge.noekeon.org/CSF-0.1.pdf, 2011.
[28] M. J. Dworkin, "Sha-3 standard: Permutation-based hash and extendable-output functions," National Institute of Standards and Technology, Tech. Rep., 2015.
[29] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST, Tech. Rep., 2010.
[30] A. I. Salih and A. A. A. Y. T., "Enhancing aes security based on dual dynamic xor table and mixcolumns transformation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1574–1581, 2020.
[31] L. H. Dinh, L. T. Thi, and L. N. Van, "On the mathematical aspects of cryptographic randomness tests using discrete fourier transform," in *2024 1st International Conference On Cryptography And Information Security (VCRIS)*. IEEE, 2024, pp. 1–6.

Tran Thi Luong received a Bachelor's degree from Hanoi University of Science, Hanoi, Vietnam, in 2006; a Master's degree in cryptographic technique at the Academy of Cryptography Techniques, Hanoi, Vietnam, in 2012; Ph.D. degree in cryptographic technique at the Academy of Cryptography Techniques, Hanoi, Vietnam in 2019. Her research interests include Cryptography, Coding theory, and Information Security.
Email: luongtran@actvn.edu.vn

Truong Minh Phuong received an Engineer of Cryptography Techniques of Academy of Cryptography Techniques in 2012; Master degreee in cryptographic technique at Academy of Cryptography Techniques in 2018. Workplace: Academy of Cryptography Techniques, No.141 Chien Thang road, Tan Trieu, Thanh Tri, Hanoi, Vietnam Recent research direction: Cryptogaphy.
Email: minhphuongh19@gmail.com