*Các công trình nghiên cứu, phát triển và ứng dụng*

**Công nghệ Thông tin và Truyền thông**

https://www.ictmag.vn/cntt-tt

**Tập 2023, Số 1, Tháng 6**

**Số đặc biệt CITA 2023**

# MỤC LỤC

*Các công trình nghiên cứu, phát triển và ứng dụng*
**Công nghệ Thông tin và Truyền thông**
https://www.ictmag.vn/cntt-tt
**Tập 2023, Số 1, Tháng 6**
**Số đặc biệt CITA 2023**

# TABLE OF CONTENT

# MIMEM: The Hybrid Metaheuristic Method for the Resource Constrained Scheduling Problem

Huu Dang Quoc[1], Loc Nguyen The[2]

[1] Thuong Mai University, 79 Ho Tung Mau, Cau Giay, Ha Noi, Viet Nam

[2] Hanoi National University of Education

Correspondence: Loc Nguyen The, locnt@hnue.edu.vn

Received: 16/04/2023, revised: 21/06/2023, accepted: 30/6/2023

Digital Object Identifier:: 10.32913/mic-ict-research-vn.v2023.n1.1218

**Abstract:** The Multi-Skill Resource-Constrained Project Scheduling Problem (MS-RCPSP) is an NP-Hard problem that involves scheduling activities while accounting for resource and technical constraints. This paper aims to present a novel hybrid algorithm called MIMEM, which combines the Memetic algorithm with the Migration method to solve the MS-RCPSP problem. The proposed algorithm utilizes the migration method to identify local extremes and then relocates the population to explore new solution spaces for further evolution. The MIMEM algorithm is evaluated on the iMOPSE benchmark dataset, and the results demonstrate that it outperforms. The solution of the MS-RCPSP problem using the MIMEM algorithm is a schedule that can be used for intelligent production planning in various industrial production fields instead of manual planning.

**Keywords:** *memetic algorithm, evolutionary computing, optimization, MS-RCPSP problem.*

---

**Tiêu đề:** **Thuật Toán Lai MIMEM Giải Bài Toán MS-RCPSP**

**Tóm tắt:** Bài toán lập lịch thực hiện dự án với tài nguyên giới hạn và đa kỹ năng (Multi Skill-Resource Constrained Project Scheduling Problem: MS- RCPSP) là một bài toán thuộc lớp NP-Hard, mục tiêu của bài toán này là tìm ra lịch biểu để thiết lập các tài nguyên thực hiện mọi công việc của dự án trong thời gian ngắn nhất. Bài báo trình bày thuật toán lai ghép mới để giải bài toán này, thuật toán mới gọi là MIMEM, là sự kết hợp giữa thuật toán Memetic với phương pháp Migration. Trong quá trình tiến hóa của thuật toán Memetic, khi quần thể rơi vào cực trị địa phương, phương pháp Migration sẽ được sử dụng để di chuyển cả quần thể sang vùng không gian nghiệm khác giúp quần thể thoát khỏi cực trị địa phương và tiếp tục quá trình tiến hóa. Để kiểm chứng tính hiệu quả của thuật toán đề xuất, bài báo đã thực nghiệm trên bộ dữ liệu iMOPSE, kết quả cho thấy MIMEM mang lại các kết quả tốt đối với bài toán MS-RCPSP. Lời giải của bài toán là lịch biểu thể hiện việc bố trí các tài nguyên thực hiện các công việc của dự án, do vậy, có thể sử dụng để lập kế hoạch điều phối sản xuất tự động thay vì các phương pháp làm truyền thống trước đây.

**Từ khóa:** *thuật toán Memetic, tính toán tiến hóa, tính toán tối ưu, bài toán MS-RCPSP*

---

## I. INTRODUCTION

The MS-RCPSP is a complex scheduling problem widely studied in operations re-search and project management. It is a type of project scheduling problem that considers the order in which tasks should be performed and the resources required to perform each task, such as labor, equipment, and materials. The goal of solving the MS-RCPSP is to minimize the project's completion time, cost, or both (multi-objective [1, 2]) while ensuring that all resource constraints are satisfied. It is a widely studied problem with numerous real-world applications, including construction, software development, and manufacturing.

The MS-RCPSP problem has an important constraint: a resource can only perform the task if it possesses the right skill type and the skill level is greater than or equal to the required skill level. Moreover, resources have many skill types and skill levels, so it is necessary to plan for allocating resources to perform practical tasks to improve the efficiency of project implementation. MS-RCPSP is a problem of class NP-Hard, so the optimal solution cannot be found in polynomial time. However, metaheuristic methods can solve the problem to find approximate solutions

in less time. In order to find a suitable solution for the MS-RCPSP problem, this paper proposes the MIMEM algorithm developed from the Memetic algorithm [3–5] integrated with the Migration method.

The rest of this paper is presented as follows: Part II introduces some research on evolutionary methods and the Memetic algorithm to solve NP-Hard problems. Part III describes the mathematical statement of the problem. Part IV presents the MIMEM algorithm to solve the MS-RCPSP problem, a hybrid algorithm between Memetic and Migration to improve the solution's efficiency by finding and detecting local extremes and moving populations away from that. Part V presents the implementation of experiments to verify the MIMEM algorithm and analyze and evaluate the quality of the solution. Finally, section VI summarizes the paper's main results and future research directions.

## II. RELATED WORKS

### 1. Approximate methods to solve the Scheduling problems

The most well-known scheduling problems are NP-Hard, meaning that their solutions cannot be found in polynomial time. Therefore, it is common to use approximate, evolutionary algorithms to find an acceptable solution quickly often use evolutionary, approximate algorithms to find an acceptable solution quickly. Many algorithms have been proposed for the scheduling problem, such as GA [8–10], PSO [11–13] , DE[14, 15], ANT [16],...

Garg et al. [17] studied and solved the scheduling problem in the cloud computing environment to improve the energy efficiency of virtual machines. The authors have proposed the PESVMC algorithm to schedule the execution of workflows. In [18], the authors study the scheduling algorithm combined with deep learning to optimize the environment that combines fog-cloud computing. The multi-objective scheduling problem is also studied and solved by parallel processing algorithms by the author of the [19] paper.

Besides, many research groups also use the GA algorithm to solve the scheduling problem. In [9], the authors use the GA integrated with the Immune to solve the NPV-Based RCPSP problem. The algorithm is also applied with two additional variables to improve efficiency: local search and forward-backward improvement. Also, using a hybrid GA algorithm to solve problems in a cloud computing environment, Hussain and colleagues [10] improved GA; this algorithm combines a blend of a dynamic round-robin and a local search algorithm and has been deployed experimentally and verified in the Cloudsim environment.

Another effective metaheuristic commonly used to solve the scheduling problem is the Particle Swarm Optimization (PSO) method. In [11], the authors studied the mod-ified PSO algorithm combined with deep learning models to apply in the image diagnosis of eye diseases. The application of modified PSO is used for both supervised and unsupervised machine learning models to tune the CLAHE parameter. In [12], PSO was studied to reduce energy consumption and the working time of stand-alone tasks in a cloud computing environment. The proposed algorithm, EE-APSO, also permits the population to eliminate the local extremes and expand the search space. In a grid computing environment, the authors use a three-PSO balanced PSO algorithm[13] to schedule workflows and improve the utilization of system resources.

Besides the above solutions, many research groups also use Differential Evolution (DE) algorithm. In [14], the authors proposed an effective method of selecting the presenter feature to improve performance and reduce the processing time of IDS systems. The proposed method uses a DE algorithm to select valuable features during an extreme machine learning classifier is applied after feature selection to evaluate the selected features. Experiments were performed on the NSL-KDD dataset. In [15], pseudo-descriptors focus on solving the individual mutation strategy to improve discoverability, convergence accuracy, and convergence speed. The authors give out a new algorithm called NBOLDE with neighborhood mutation and opposition-based learning operators. The proposed NBOLDE includes new evaluation parameters and weighting factors in the neighborhood model to propose a new neighborhood strate-gy. On this basis, a new neighbor mutation strategy based on DE to best. Usually, to experiment with the proposed algorithms, the authors often use two primary datasets, PSLIB[20] and iMOPSE [6, 7], which P.B. Myszkowski suggested

### 2. Memetic algorithm

Memetic algorithms (MAs)[3–5] are optimization algorithms that incorporate both evolutionary and local search procedures to solve complex optimization problems. MAs have gained increasing popularity in recent years due to their ability to over-come the limitations of traditional evolutionary algorithms (EAs) by incorporating problem-specific knowledge to guide the search process.

In a memetic algorithm, individuals in the population are encoded as solutions to the optimization problem and evolve through selection, crossover (recombination), and mutation. The genetic algorithm generates new candidate solutions, while the local search method is used to refine the solutions generated by the genetic algorithm.

Using local search methods in memetic algorithms can lead to faster convergence and improved solution quality compared to traditional genetic algorithms. This is because local search methods can exploit the structure of the problem space to find better solutions quickly. At the same time, the genetic algorithm provides a mecha-nism for exploring the search space and avoiding getting trapped in local optima.

However, there are also some disadvantages to using MAs. One of the main draw-backs is that MAs can be computationally expensive, as they require both global and local search procedures. Another challenge is designing effective, problem-specific, and efficient local search procedures can be complex. Furthermore, MAs can also suffer from premature convergence, where the algorithm becomes stuck in a local optimum and cannot escape finding the global optimum.

## III. MS-RCPSP PROBLEM

The MS-RCPSP has a wide range of real-world applications, such as in project management, scheduling of manufacturing processes, and resource allocation in large-scale engineering projects. The ability to effectively solve the MS-RCPSP has significant implications for the efficiency and success of these real-world applications.

The MS-RCPSP problem can be conceptually formulated based on the notations in Table I.

The MS-RCPSP problem could be state as follow:

$$f(P) \rightarrow min \tag{1}$$

Subject to the following constraints:

$$S^k \neq \oslash \quad \forall L_k \in L \tag{2}$$

$$t_j \geq 0 \quad \forall W_j \in W \tag{3}$$

$$E_j \geq 0 \quad \forall W_j \in W \tag{4}$$

$$E_i \leq E_j - t_j \quad \forall W_j \in W, j \neq 1, W_i \in C_j \tag{5}$$

$$\forall W_i \in W^k \quad \exists S_q \in S^k : g_(S_q) = g_(r^i) \text{ và } h_(S_q) \geq h_(r^i) \tag{6}$$

$$\forall L_k \in L, \forall q \in m : \sum_{i=1}^{n} A_{i,k}^q \leq 1 \tag{7}$$

$$\forall W_j \in W \exists! q \in [0-m], !L_k \in L : A_{j,k}^q = 1; \ |A_{j,k}^q \in \{0;1\} \tag{8}$$

In the MS-RCPSP problem, each task has additional skill requirements of the renew-able resource needed to perform. Each resource is also divided into different skill types and skill levels. Figure 1 depicts an example of the resource skill constraints required to complete a task.

## IV. PROPOSED ALGORITHM

This section describes the proposed evolutionary algorithm for the MS-RCPSP prob-lem named MIMEM, a variation of the Memetic algorithm [3–5] enhanced with the Migration method.

In the MS-RCPSP problem, the population is calculated and evolved over several generations to improve the makespan of the project. Representing each individual in the population is a vector whose elements correspond to the total number of project tasks. The value at each vector element denotes the resource index assigned to perform the corresponding task.

### 1. Migration method

The traditional memetic algorithm operates by evolving a population over several generations through several steps, including selection, crossover, mutation, local search, and recombination to create a new population. However, while evolving, the algorithms may fall into the local extreme and iterate infinitely without leaving, so finding a better result is impossible. Therefore, the Migration technique moves the population to another solution space far from the local extreme. Integrating the Migration technique into the memetic algorithm leads to improved results.
The Migration method runs over steps as follows:

*Step 1: Catching the local extreme*

In this step, the algorithm uses a marked variable $c_{fail}$ to count the times the makespan is not changed over generations; if $c_{fail}$ was more significant than a specified threshold ($c_{max}$), then the population has fallen into the local extreme. Equation (10) represents the calculation of $c_{fail}$. (10)

*Step 2: Changing the population*

Once the population falls into a local extreme, the MIMEM algorithm tries to move the entire population to a new area by using the Migration method as follows:

- Repeating with each particle
- Considering each task $i$ of the particle, get out the $L_i$ of resources to execute the current task, reordering the resources index in $L_i$.

Table I
THE NOTATIONS

| Symbol | Description |
|---|---|
| $C_i$ | The set of tasks need to be completed before task $i$ can be executed |
| $S$ | The set of all resource's skills $S_i$: the subset of skills owned by the resource $i$, $S^i \subseteq S$; |
| $S_i$ | The skill $i$; |
| $t_j$ | The duration of task $j$ |
| $L$ | The resources used to execute tasks of the project |
| $L^k$ | The subset of the resources which can be performed task $k$; $L^k \subseteq L$ |
| $L_i$ | The resource $i$ |
| $W$ | The tasks of the project need to do |
| $W^k$ | The subset of task which can be executed by the resource $k$ , $W^k \subseteq W$ |
| $W_i$ | The task i |
| $r^i$ | The subset of the skill required by task $i$. A resource has the same skill and skill level equal to or greater than the requirement that can be performed. |
| $B_k, E_k$ | The begin time and end time of the task $k$ |
| $Au, vt$ | The variable to identify the resource $v$ is running task $u$ at time $t$; *1*: yes, *0*: no; |
| $hi$ | The skill level $i$; |
| $gi$ | Type of skill $i$; |
| $m$ | Makespan of the schedule |
| $P$ | The feasible solution |
| $P_{all}$ | The set of all solution |
| $f(P)$: | The function to calculate the makespan of $P$ solution |
| $n$ | Task number |
| $z$ | Resource number |

- Replacing the current resource with the resource which mirrors in $L_i$. The formula (11) shows this action.

Figure 1 presents the Migration technical by replacing the execution resource. Primarily, the task being performed by resource $L_2$ will be changed to $L_{m-1}$, and resource $L_m$ does the task will be allocated to $L_1$.

After applying the Migration method, the population will be moved to a new location, allowing the algorithm MIMEM to escape the local extreme, continue the evolution-ary computation, and expand the solution space.

The Migration method is present in Algorithm 1 below.

## 2. MIMEM Algorithm

The MIMEM algorithm is a hybrid algorithm from the memetic algorithm and the Migration method. The improvement of MIMEM is demonstrated in "Migration steps", where the algorithm checks for local extrema within the population and redirects it to another region if it has fallen into one. This step is executed by applying the Migration method.

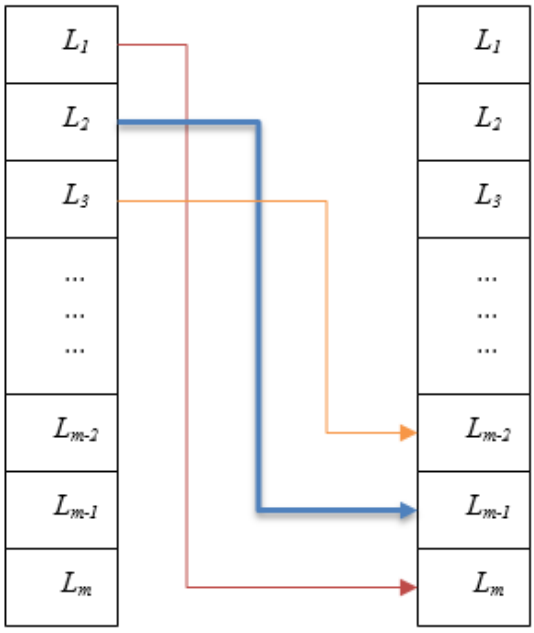



Figure 1. Replace the resource that performs the task

**Algorithm 1:** Migration

```
1  Input: dataset, g_max: number of generation, mutation-
     Probability, crossoverProbability, localSearchProbability
2  Output: best and duration
3  begin
4      P_new = {};
5      i = 1;
6      p_size = size(P);
7      while i < p_size do
8          P_i ← P_all[i];
9          j_task = 1;
10         n_task = size(P_i);
11         for j_task = 1 to n_task do
12             L^i ← the resources matching with L(j_task)
                 requirement;
13             L^i ← Reorder(L^i) ;
14             i_current ← current resource index to run
                 task i;
15             i_current = size(L^i) – idx + 1;
16             L_jtask = L^i[i_current];
17         end
18         P_new = P_new + {P_i};
19     end
20     return P_new;
21 end
```

**Algorithm 2:** MIMEM

```
1  Input: P_all – the population needs to move
2  Output: P_new – the result population
3  begin
4      Read dataset;
5      g = 0, c_fail = 0, c_max = 50;
6      P_all = {population initallation} ;
7      {makespan, best, fitness} = {Caculating the fitness of
         population and the best individual} ;
8      while g < g_max do
9          parents = Selection(population);
10         offspring = Crossover(population, crossoverProb-
             ability);
11         Mutate(offspring, mutationProbability);
12         LocalSearch(offspring, localSearchProbability);
13         P_all = CombinePopulation(P_all, offspring);
14         {new_makespan, new_best, fitness} = {Cacu-
             lating the fitness of population and the best
             individual from P_all} ;
15         if new_makespan # makespan then
16             c_fail = 0 ;
17         else
18             c_fail+ = 1;
19         end
20         if c_fail = c_max then
21             P_all = Migration(P_all);
22             c_fail = 0;
23         end
24         if makespan < new_makespan then
25             makespan = new_makespan; ;
26             best = new_best ;
27         end
28         g = g+1 ;
29     end
30     return {best, makespan } ;
31 end
```

The MIMEM algorithm is present in Algorithm 2.

At each evolution generation of the population, the local search technique is performed by calling the LocalSearch function, which relies on the localSearchProbability parameter to calculate the number of neighbor individuals ($localSearchProbability * sizeof(P_{all})$) which they are used to be calculated to find the best one and return it. The best individual among the neighbors has been selected if a better makespan exists than the current.

In Algorithm 2, lines 15-19 aim to detect local extremes, while commands in lines 20-23 move the population by calling the Migration function. After moving the population, the algorithm continues on the new solution area.

## V. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed MIMEM algorithm, we exper-imented on the iMOPSE [6, 7] dataset presented in Table II.

### 1. Experimental parameters

Experiments were conducted with the following parameters and data:

- Benchmark instances: iMOPSE dataset (iM01 to iM15), each instance is a project with full input parameters such as the number of tasks, the number of resources used to execute the project, the number of precedence constraints, and the number of resource skills.
- The initial number of particles (i.e., solution, individual, schedule): 50
- The number of generations: 30,000
- The number of times the experiment was carried out on each element of the dataset: 30
- Experiment tools: Matlab version 2014
- PC configuration: CPU Core i5 2.2 GHz, 6GB RAM, Windows 10

### 2. Performance Analysis

Experimental results with the iMOPSE dataset are presented in Table III. The results are aggregated, compared, and evaluated based on three factors:

- BEST - best makespan values - this is the shortest time to complete the project
- AVG - average execution time of 50 experiments with each instance dataset
- STD values - the standard deviation value between experiments on the same data instance

Table II
THE IMOPSE DATASET

| Name | Tasks | Resources | Precedence Relations | Skills |
|---|---|---|---|---|
| iM01 | 100 | 5 | 22 | 15 |
| iM02 | 100 | 5 | 46 | 15 |
| iM03 | 100 | 5 | 48 | 9 |
| iM04 | 100 | 5 | 64 | 15 |
| iM05 | 100 | 5 | 64 | 9 |
| iM06 | 100 | 10 | 26 | 15 |
| iM07 | 100 | 10 | 47 | 9 |
| iM08 | 100 | 10 | 48 | 15 |
| iM09 | 100 | 10 | 64 | 9 |
| iM10 | 100 | 10 | 65 | 15 |
| iM11 | 100 | 20 | 22 | 15 |
| iM12 | 100 | 20 | 46 | 15 |
| iM13 | 100 | 20 | 47 | 9 |
| iM14 | 100 | 20 | 65 | 15 |
| iM15 | 100 | 20 | 65 | 9 |

The average execution time of the GA algorithm and the MIMEM algorithm are 2.7 hours and 2.3 hours, respectively. Note that normally the time it takes to find a schedule makes almost no sense because the quality of its solution is what matters.

Myszkowski has published GArunner tool [21] that allows other research groups to implement his GA algorithm, which is considered one of the most efficient algorithm. Thanks to that tool, we implemented the GA algorithm and the MIMEM on the iMOPSE dataset [6, 7]. The results are shown in Table III.

The experimental results show that the proposed algorithm MIMEM is more efficient than the GA algorithm on the following parameters:

- In term of BEST values, the MIMEM algorithm outperforms GA in 12 out of 15 projects, achieving improvements ranging from 0 to 22.33%. However, in 3 projects, MIMEM performs worse than GA. Figure 2 illustrate the BEST values comparisons detail.
- Regarding AVG values, MIMEM yields better results than GA in 11 out of 15 projects, achieving improvements ranging from 0 to 23.31%. However, in 4 projects, MIMEM performs worse than GA. Figure 3 detail the AVG values comparisons for 15 projects.
- Moreover, MIMEM exhibits greater stability than GA with a total standard deviation of 49.8, compared to GA's 68.8 on 15 datasets of iMOPSE. Figure 4 present the STD values comparisons for 15 projects.

Experimental results demonstrate that the MIMEM algorithm is less effective for projects with fewer tasks, but it achieves higher efficiency for larger projects. This suggests that MIMEM is well-suited for scheduling real-world projects consisting of a series of jobs with a significant number of tasks. For instance, MIMEM can be applied in various fields, such as equipment manufacturing and industrial sewing lines,...

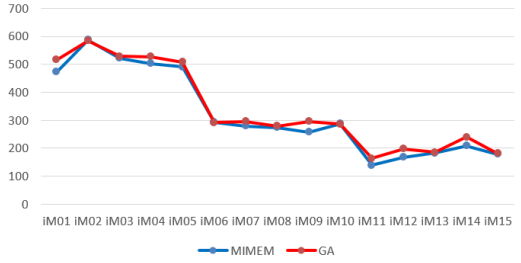



Figure 2. Comparison of the BEST parameter between MIMEM and GA

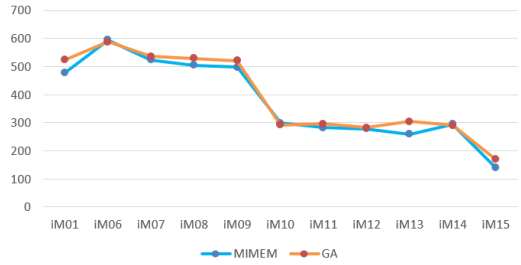



Figure 3. Comparison of the AVG parameter between MIMEM and GA

Table III
THE RESULTS FROM THE IMOPSE DATASET

| Dataset | GA | | | MIMEM | | |
|---|---|---|---|---|---|---|
| | **Best** | **Avg** | **Std** | **Best** | **Avg** | **Std** |
| iM01 | 517 | 524 | 5.3 | 473 | 477 | 2.5 |
| iM02 | 584 | 587 | 4.7 | 587 | 594 | 4.3 |
| iM03 | 528 | 535 | 9.7 | 522 | 523 | 10.4 |
| iM04 | 527 | 530 | 2.5 | 503 | 504 | 1.6 |
| iM05 | 508 | 521 | 9.9 | 490 | 497 | 4.7 |
| iM06 | 292 | 292 | 1.7 | 294 | 299 | 0.1 |
| iM07 | 296 | 296 | 2.3 | 280 | 283 | 1.7 |
| iM08 | 279 | 282 | 2.9 | 274 | 278 | 3.4 |
| iM09 | 296 | 305 | 6.6 | 258 | 259 | 0.7 |
| iM10 | 286 | 290 | 5.0 | 287 | 294 | 3.5 |
| iM11 | 163 | 169 | 5.8 | 139 | 139 | 1.9 |
| iM12 | 197 | 207 | 6.9 | 168 | 169 | 8.2 |
| iM13 | 185 | 186 | 0.5 | 183 | 188 | 0.4 |
| iM14 | 240 | 240 | 0.5 | 208 | 211 | 5.3 |
| iM15 | 181 | 187 | 4.5 | 179 | 185 | 1.1 |

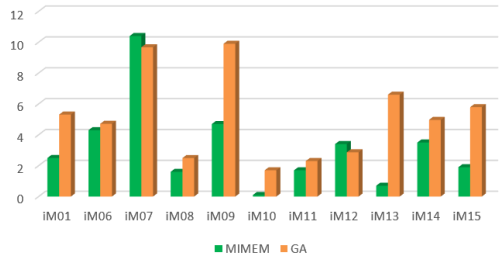



Figure 4. Comparison of the STD parameter between MIMEM and GA

## VI. CONCLUSION

In this paper, we have presented a mathematical formulation of the MS-RCPSP problem and proposed a new algorithm, MIMEM, that combines the strengths of the Memetic algorithm and the Migration method. Our experiments on the IMOPSE dataset demonstrate that MIMEM outperforms previous algorithms regarding BEST and AVG values, achieving up to a 22.33% improvement on BEST values and a 23.31% improvement on AVG values. The total STD value suggests that the pro-posed algorithm exhibits more excellent stability, meaning that it produces less difference between experiment times.

The results suggest that MIMEM is a promising approach for solving complex scheduling problems with a large number of tasks, which is critical for various real-world industrial production scenarios such as industrial machine lines or some other manufacturing sectors. In future work, we plan to investigate other approximation methods, such as random moves based on Gauss, Cauchy, etc., to enhance further the quality of the schedules produced by MIMEM.

## REFERENCES

[1] Zhang, Xin, et al. "A coevolutionary algorithm based on the auxiliary population for con-strained large-scale multi-objective supply chain network.", *Mathematical Biosciences and En-gineering*, 19.1 (2022): 271-286.

[2] Zhang, Wenqiang, et al. "Multi-stage hybrid evolutionary algorithm for multiobjective distrib-uted fuzzy flow-shop scheduling problem.", *Mathematical Biosciences and Engineering*, 20.3 (2023): 4838-4864.

[3] Wilson, Allan J., et al. "A review on memetic algorithms and its developments.", *Electrical and Automation Engineering*, 1.1 (2022): 7-12.4.

[4] Afsar, Sezin, et al. "Multi-objective enhanced memetic algorithm for green job shop schedul-ing with uncertain times.", *Swarm and Evolutionary Computation*, 68 (2022): 101016.

[5] Seo, Wangduk, et al. "Effective memetic algorithm for multilabel feature selection using hy-bridization-based communication.", *Expert Systems with Applications*, 201 (2022): 117064.

[6] Myszkowski, Paweł B., and Maciej Laszczyk. "Investigation of benchmark dataset for many-objective Multi-Skill Resource Constrained Project Scheduling Problem.", *Applied Soft Computing*, 127 (2022): 109253.

[7] P.B. Myszkowski, M. Laszczyk, I. Nikulin, M. Skowro, "iMOPSE: a library for bicriteria op-timization in Multi-Skill Resource-Constrained Project Scheduling Problem", *Soft Computing Journal*, 23: 32397, 2019.

[8] Saad, Hatem MH, Ripon K. Chakrabortty, Saber Elsayed, and Michael J. Ryan. "Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling.", *IEEE Access*, 9 (2021): 38488-38502.

[9] Asadujjaman, Md, Humyun Fuad Rahman, Ripon K. Chakrabortty, and Michael J. Ryan. "An Immune Genetic

Algorithm for Solving NPV-Based Resource Constrained Project Schedul-ing Problem.", *IEEE Access*, 9 (2021): 26177-26195.
[10] Hussain, Adedoyin A., and Fadi Al-Turjman. "Hybrid Genetic Algorithm for IOMT-Cloud Task Scheduling.", *Wireless Communications and Mobile Computing*, 2022 (2022).
[11] Aurangzeb, Khursheed, Sheraz Aslam, Musaed Alhussein, Rizwan Ali Naqvi, Muhammad Arsalan, and Syed Irtaza Haider. "Contrast Enhancement of Fundus Images by Employing Modified PSO for Improving the Performance of Deep Learning Models.", *IEEE Access*, 9 (2021): 47930-47945.
[12] Rani, Rama, and Ritu Garg. "Energy efficient task scheduling using adaptive PSO for cloud computing.", *International Journal of Reasoning-based Intelligent Systems*, 13.2 (2021): 50-58.
[13] Sahana, Sudip Kumar. "Ba-PSO: A Balanced PSO to solve multi-objective grid scheduling problem.", *Applied Intelligence*, (2021): 1-13.
[14] Al-Yaseen, Wathiq Laftah, Ali Kadhum Idrees, and Faezah Hamad Almasoudy. "Wrapper feature selection method based differential evolution and extreme learning machine for intru-sion detection system.", *Pattern Recognition*, 132 (2022): 108912.
[15] Deng, Wu, et al. "An improved differential evolution algorithm and its application in optimiza-tion problem.", *Soft Computing*, 25.7 (2021): 5277-5298.
[16] Zhao, Suyao, et al. "Large-Scale Scheduling Model Based on Improved Ant Colony Algo-rithm." *Mobile Information Systems* 2022 (2022).
[17] Garg, Neha, et al. "Energy-Efficient Scientific Workflow Scheduling Algorithm in Cloud En-vironment.", *Wireless Communications and Mobile Computing* 2022 (2022).
[18] Shruthi, G., et al. "Mayfly taylor optimisation-based scheduling algorithm with deep rein-forcement learning for dynamic scheduling in fog-cloud computing.", *Applied Computational Intelligence and Soft Computing* 2022 (2022).
[19] Shams Lahroudi, Seyed Hassan, Farzaneh Mahalleh, and Seyedsaeid Mirkamali. "Multiobjec-tive Parallel Algorithms for Solving Biobjective Open Shop Scheduling Problem.", *Complexity* 2022 (2022).
[20] R. Kolisch, A. Sprecher, "PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program.", *European journal of operational research*, 96(1), pp.205-216, 1997.
[21] GArunner tool: http://imopse.ii.pwr.wroc.pl/rcpsp_spsp _library.html

**Huu Dang Quoc** received Bachelor and M.S. degree in School of Information Technology, Vietnam National University, Ha Noi, Viet Nam, in 2000 and 2015. He received Ph.D. degree in Military Institute of Science and Technology, Ha Noi, Vietnam, Viet Nam, 2022. He worked at Thuong Mai University, Ha Noi, Viet Nam from 2006. His research interests include Software Engineering, Evolution Algorithm, Optimization Algorithm.
E-mail: huudq@tmu.edu.vn

**Loc Nguyen The** received Bachelor and M.S. degree in School of Information and Communication Technology, Hanoi University of Science and Technology, Viet Nam, in 1998 and 2001, respectively. He received Ph.D. degree in School of Information Science, Japan Advanced Institute of Science and Technology, Japan, 2007. He worked at Hanoi National University of Education, Viet Nam from 1997 and is currently a professor, vice dean of the Department of Information Technology. His research interests include Computer Network and Computer.
E-mail: locnt@hnue.edu.vn.

# A Post-Filtering for Performance of Differential Microphone Array

Nguyen Ba Huy[1], Pham Tuan Anh[2], Quan Trong The[3]
[1]Falculty of Control Systems and Robotics, University ITMO St.Petersburg, Russia
[2]Vietnam - Korea University of Information and Communication Technology, The University of Danang, Vietnam
[3]Falculty of Software Engineering and Computer Systems, University ITMO St.Petersburg, Russia

Correspondence: Quản Trọng Thế, Email: quantrongthe1984@gmail.com
Received: 13/03/2023, revised: 21/06/2023, accepted: 25/06/2023
Digital Object Identifier:: 10.32913/mic-ict-research-vn.v2023.n1.1221

**Abstract:** Microphone array (MA) is a perspective technique for speech enhancement, because of its simplicity and low computation. MA can be decomposed into a concatenation of a spatial filtering, which provides beampattern toward a certain target directional speech source while attenuating noise, interference or third-party speaker from other directions, and a post-filtering, which suppresses the remaining noise at the final output signal. Differential Microphone Array (DIF) is one basic MA, which has many advantages for extracting the desired speech and null-steering to direction of noise. Nevertheless, in real situation, the performance of DIF often deteriorated due to phase error, imprecise the Direction-Of-Arrival (DOA) of interest signal, microphone mismatches or imperfect distribution of MA, which significantly degrade the performance and the speech quality of DIF. In this article, the author proposed a Post-Filtering (PF), which based on the properties of considered recording environment DOA to enhance the speech quality of target speaker and reduce the other non-target speaker at the opposite direction. Experimental results were verified in real scenario with dialogue between two speakers, which stand opposite direction, and show the effectiveness of the suggested PF when compared with the author's previous work.

**Keywords:** *microphone array signal, noise reduction, post-filtering, the direction of arrival, speech enhancement*

---

**Tiêu đề:** **Một phương pháp Post-Filtering khi áp dụng lưới microphone vi sai**

**Tóm tắt:** Công nghệ lưới microphone là một trong những kỹ thuật tối ưu để nâng cao chất lượng thoại, bởi tính đơn giản, mức độ phép tính thấp. Lưới microphone thường được hiểu là phép lọc không gian, trong đó cung cấp giản đồ đáp ứng xung hướng về một nguồn thoại cụ thể, trên hướng xác định trong khi gây suy hao nhiễu, người nói bên ngoài từ các hướng khác, và kỹ thuật Post- Filtering, cho phép triệt tiêu thành phần nhiễu còn lại tại tín hiệu đầu ra. Lưới microphone vi sai cũng là một dạng cơ bản, có một số ưu điểm về tách lọc tín hiệu thoại gốc và triệt giảm nhiễu xung quanh. Tuy nhiên, trong những tình huống thực tế, kết quả của lưới microphoe vi sai thường bị suy giảm, cho sai số về pha, sự không chính xác về góc tới tín hiệu thoại gốc, mất dữ liệu khi thu ghi hoặc lưới microphone phân bố không đều, điều này có ảnh hưởng rất lớn đến hiệu quả, và chất lượng tín hiệu thoại cuối cùng. Trong bài nghiên cứu này, nhóm tác giả đề xuất một Post-Filtering mới, dựa trên đặc tính hướng góc tới môi trường thu ghi để nâng cao chất lượng thoại mục tiêu trong khi triệt giảm thành phần thoại của người nói hướng đối diện. Kết quả thực nghiệm đã được chứng minh trong môi trường thu ghi thực tế trao đổi thoại giữa hai người nói đối diện nhau, và chỉ rõ hiệu quả của Post-Filtering đề xuất khi so sánh với nghiên cứu trước đây của tác giả.

**Từ khóa:** *lưới microphone, giảm nhiễu, post-filtering, hướng góc tín hiệu tới, nâng cao chất lượng thoại.*

---

## I. INTRODUCTION

Almost single-channel algorithms are based on the spectral subtraction method, which often leads to speech distortion and decreasing the speech intelligibility and quality of the output signal. Therefore, MA [1-8] is one of the most effective techniques for solving this coherent drawback. Acoustic digital processing devices such as a mobile telephone, cochlear implant, hearing aid, audio device is widely equipped with an MA to capture the acoustic environment, target directional speaker, transport vehicle, interference, and surrounding noise. A mixture of speech plus some complex noisy signals, reverberation, background interference, third-party speaker, imperfect propagation of sound source

or microphone mismatch are often occurred with signal processing. Therefore, the task of alleviating background noise and extracting the target speech speaker, known as a critical importance speech enhancement, has been an attractive subject of extensive direction of improvement. Intelligibility and quality of the final output signal is a major factor when estimating a speech system in presence of such environmental noise.

Due to the exploiting the spatial filtering, the knowledge of target speech source, non-target noise, the characteristic of recording situation, MA technology are successfully integrated into various speech enhancement scenarios. However, because of the constraint of physical size or the distance between microphones or geometry of MA, an acceptance degree of extracting the only desired speaker without speech distortion cannot be yielded although there are lot of sufficient spatial beamforming between the observed real microphone arrays data. Consequently, a post-filtering is enough to reduce the remaining noise after a beamformer and increase the speech quality and intelligibility.

DIF is one of the basic elements of MA, which allows blocking a point of noise source by forming a null-beampattern in the direction of noise. In the previous author's paper [26], and equalizer is applied to enhance the DIF's performance in situation of two opposite speaker. However, in many speech applications, the necessary requirement is yielding the only target speaker without noise, interference, or the other different speaker. So, in this article, the author proposed a post-filtering, which is based on the different phase to improve speech enhancement by suppressing the remaining speech component of non-target speakers.

In many decades, a numerous works focused on the designing of compact and small-size DMA [4-8], due to their low computational cost, approximately perfect invariant-frequency beampattern and good directivity index. Spatial diversity of DMAs can be calculated and optimized in any arbitrary order of beampattern, especially first-order are preferred widely commonly utilized in realistic practical applications. In real-life environment, DMAs are affected by many types of noise: white noise, diffuse noise, coherent and incoherent noise, and has many characteristics: phase mismatches, high sensitivity to misplacements, or error of omnidirectional microphones, amplification in low frequency.

The interesting term in signal processing technology is achievement of directional beampattern for compact and small-size MA, which has been an importance require due to the proliferation of embedded system for mobile phone, acoustic surveillance, infotainment. The small-size arrays, as the distance between microphones must be small enough to satisfy the assumption that the directional beampattern can be steered toward target speech. The smaller distance, the wider the spectrum ranges in obtaining beampattern with perfect nearly invariant-frequency characteristic. Moreover, the polar pattern of DMA can be adjusted by varying the value of delay time in microphone array signals. For example, Uniform Linear Arrays (ULAs) and Non-Uniform Linear Arrays (NULAs) allow receiving the symmetric beam with respect to the axis of MA.

The development of strategies for designing steerable beampattern with high order to any arbitrary direction has been attractive to scholars. In this approach, Elko [9-12] proposed a method, which is based on a linear combination of eigen-beam shaped of different order DMA, one direction focused on target talker and other directions toward noise source. Otherwise, many scholars developed designs to receive high robustness, noise reduction, directivity index and flexibility of steering pattern in any arbitrary configuration of MA or distribution of microphones [13-20]. These works have been evaluated and verified the effective performance in extracting target directional speech while suppressing all background noise and increased the robustness of DMA's working.

An essential important issue is the steering flexibility in practical applications with any arbitrary type of MA: linear, circular, planar MA. Numerous efforts have been proposed to enhance the steering flexibility of DMAs [21-25]. With great development has been obtained in the design of DMAs with good steerable pattern, the capability of DMA becomes more importance in microphone arrays signal processing.

Dual-microphone system (DMA2) is the most popular model MA is commonly installed in almost acoustic device, because of it's compact, it's convenient to easy implement almost MA processing algorithm. In experiment, DMA2 is used for recording two speakers and illustrating the proposed post-filtering in comparison with the previous author's work.

This contribution is organized as follows. In the section II, an analysis of principle working of DIF is described, and the model of equalizer, which is used in [26] is presented. The suggested post-filtering is described in Section III. Section IV and V is experiment and Conclusion.

## II. DIFFERENTIAL MICROPHONE ARRAY

DMA2 helps achieving good steering flexibility, high noise reduction, high directivity index, a suitable directional pattern, which towards to target speech source. Due to its small size, low computational cost and easy to implement, DMA is the most appropriate structure for extracting desired speech source and eliminating background noise.

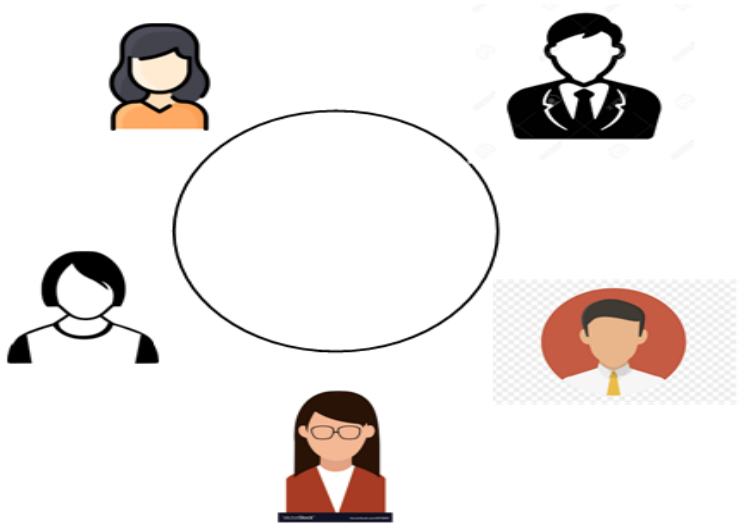


Figure 1. The most difficult task in speech enhancement is speech separation.

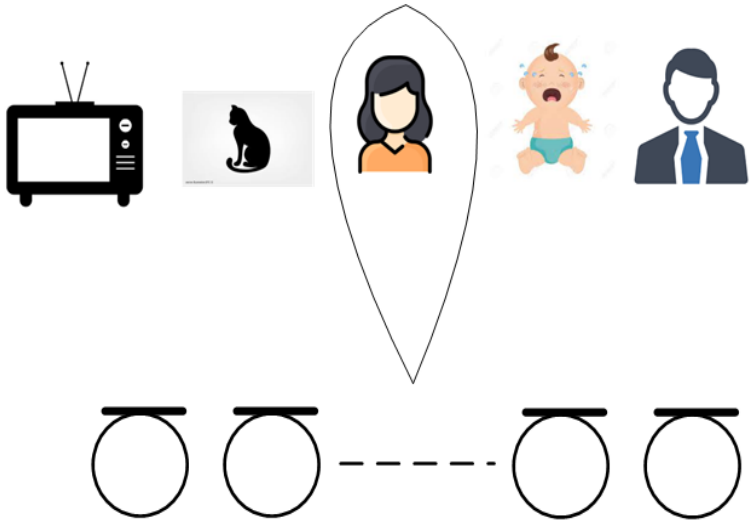


Figure 2. The using of MA to extract the desired target directional speaker.

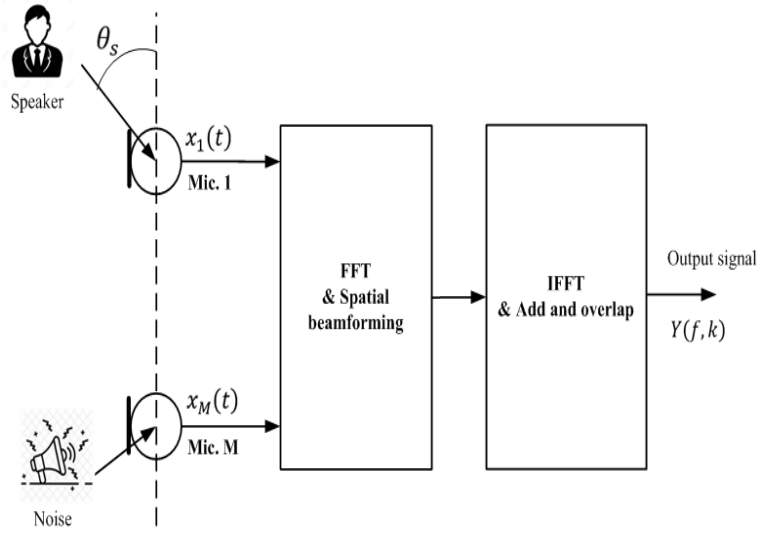



Figure 3. The scheme of MA to process microphone array signals.

We consider the principle DMA2's working in the STFT domain.

Two noisy microphone arrays signals $X_1(f,k), X_2(f,k)$ are represented in STFT domain as:

$$X_1(f,k) = S(f,k)e^{j\Phi_s} \quad (1)$$

$$X_2(f,k) = S(f,k)e^{-j\Phi_s} \quad (2)$$

Where $\theta$ is the direction-of-arrival of useful signal, $S(f,k)$ is the target desired speech source, $\Phi_s = \pi f \tau_0 cos\theta$, $\tau_0 = d/c$, $d$ is the inter-distance between two microphones, $c = 343(m/s)$ is the speech of sound propagation; $f, k$ is the index of frequency and frame, respectively.

With $\tau$: an appropriate delay time was added, the high flexible steerable directivity pattern of DMA2 obtained by the described equations:

$$Y_{1DIF}(f,k) = \frac{X_1(f,k) - X_2(f,k)e^{-j\omega\tau}}{2} \quad (3)$$

$$= jS(f,k)sin(\frac{\omega\tau_0}{2}(cos\theta + \frac{\tau}{\tau_0})) \quad (4)$$

$$Y_{2DIF}(f,k) = \frac{X_2(f,k) - X_1(f,k)e^{-j\omega\tau}}{2} \quad (5)$$

$$= -jS(f,k)sin(\frac{\omega\tau_0}{2}(cos\theta - \frac{\tau}{\tau_0})) \quad (6)$$

The signal subtraction operation allows deriving equation of directional beampattern. These patterns can be expressed as:

$$B_1(f,\theta) = |\frac{Y_1(f,k)}{S(f,k)}| = |sin(\frac{\omega\tau_0}{2}(cos\theta + \frac{\tau}{\tau_0}))| \quad (7)$$

$$B_2(f,\theta) = |\frac{Y_2(f,k)}{S(f,k)}| = |sin(\frac{\omega\tau_0}{2}(cos\theta - \frac{\tau}{\tau_0}))| \quad (8)$$

In the previous work, a suggested Equalizer was used for deriving the original speech component of talker. And the Equalizer can be expressed as:

$$H_{eq}(f) = \begin{cases} 6 & 0 \text{ Hz} < \text{f} \leq 200 \text{ (Hz)} \\ \frac{1}{\sin(\frac{\pi}{2}\frac{f}{Fc})} & 200 \text{ Hz} < \text{f} \leq \text{Fc} \\ 1 & \text{Fc} < \text{f} \leq 2\text{*Fc} \\ 0 & 2\text{*Fc} < \text{f} \end{cases} \quad (9)$$

where $Fc = \frac{1}{4*\tau_0}$.

A constrained constant threshold 12 (dB) was set for $H_{eq}(f)$ to avoid gaining the amplitude of original signal.

The obtained final output signals as:

$$Y_1(f,k) = Y_{1DIF}(f,k) * H_{eq}(f) \quad (10)$$

$$Y_2(f,k) = Y_{2DIF}(f,k) * H_{eq}(f) \quad (11)$$

## III. THE PROPOSED POST-FILTERING

We can easily determine two additive post-filtering, which use the information about directions. With two preferred directions $\theta_1 = 0(deg), \theta_2 = 180(deg)$, the authors proposed exploiting the different phase $\theta_k(f)$ between two microphone array signals. $\theta_k(f)$ can be calculated as the following way in the frequency domain:

$$\theta_{k1}(f) = \arg(X_1(f,k)) - \arg(X_2(f,k)) - 2\pi f \tau_0 cos\theta_1 \quad (12)$$

$$\theta_{k2}(f) = \arg(X_1(f,k)) - \arg(X_2(f,k)) - 2\pi f \tau_0 cos\theta_2 \quad (13)$$

$\theta_{k1}(f), \theta_{k2}(f)$ is always in range $[-\pi..\pi]$. In the frames, which contain the speech component, $\theta_{k1}(f), \theta_{k2}(f)$ tends to 0 and in the noisy frame, $\theta_{k1}(f), \theta_{k2}(f)$ tends to $-\pi$ or $\pi$.

An effective Post-Filtering, which uses $\theta_k(f)$ as:

$$H_{Post-Filtering1}(f) = \cos\frac{\theta_{k1}(f)}{2} \quad (14)$$

$$H_{Post-Filtering2}(f) = \cos\frac{\theta_{k2}(f)}{2} \quad (15)$$

At the frames, in which the desired target speaker exits, $H_{Post-Filtering1}(f), H_{Post-Filtering2}(f)$ equal approximately 1 and will save the speech component. Thus in, at the other frames, $H_{Post-Filtering1}(f), H_{Post-Filtering2}(f)$ close to 0 and remove the different speech component of third-party speaker.

And the final enhanced output signals can be yielded by:

$$Y_{1pro}(f,k) = Y_1(f,k) * H_{Post-Filtering1}(f) \quad (16)$$

$$Y_{2pro}(f,k) = Y_2(f,k) * H_{Post-Filtering2}(f) \quad (17)$$

## IV. EXPERIMENTS

In this section, the authors illustrated an perspective experiment with two speakers and a dual-microphone system. The speaker stand at distance $L = 2(m)$, the inter-microphone distance $d = 5(cm)$, two speakers talk with each other at opposite direction. For further signal processing, the sampling rate is $Fs = 16kHz$, overlap 50%, smoothing parameter $\alpha = 0,1$, a Hamming window were used for transforming received arrays signals into STFT domain. The scheme of experiment is described in Figure 6. In this experiment, DMA2 recorded speech component from target directional speaker along the axis of DMA2, $\theta_s = 0(deg)$; while at the opposite direction $\theta_v = 180(deg)$, a noise source corrupted the speech quality. The purpose of experiment is illustrating the capability of suppressing

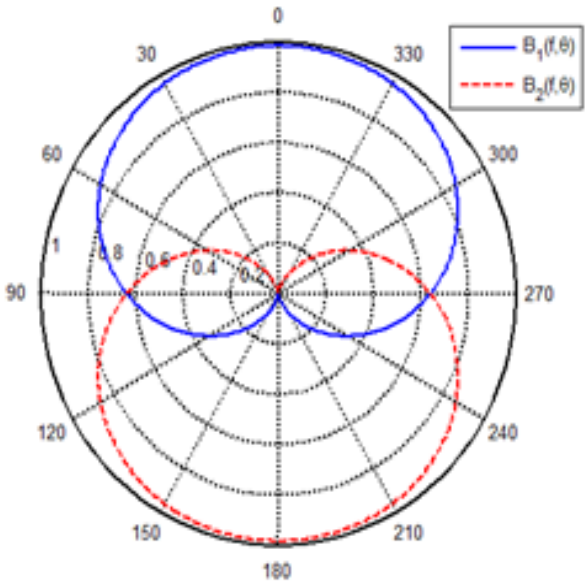



Figure 4. The desired beampattern with DMA2, $d = 5(cm)$, $f = 1500(Hz)$.

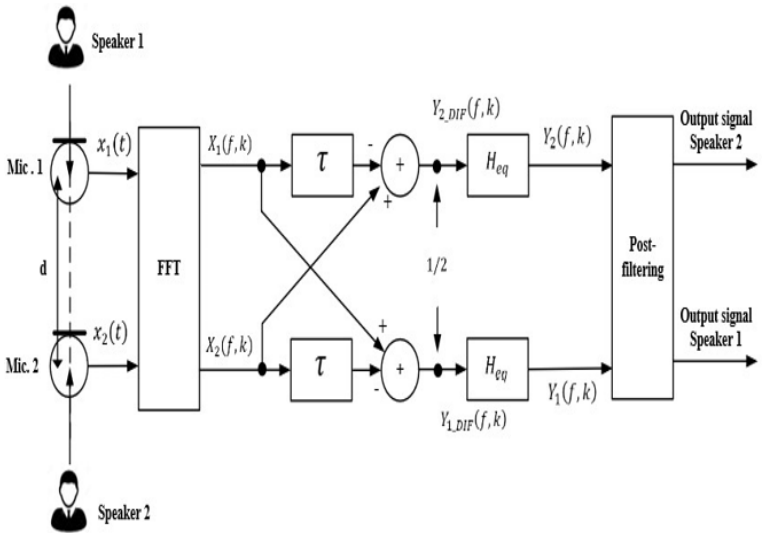



Figure 5. The model of DMA2 for extracting useful speech source.

the remaining second speaker's speech component of second speaker while saving first speaker's target speech in comparison with the previous author's work.

The original microphone array signal is presented in Figure 7.

The processed signal by the proposed method and the previous author's work [26] are depicted in Figure 8. The first speaker's speech component is saved, while the second speaker's component still exist with a substantial amount of speech component.

From figure 9, the effectiveness of the suggested post-filtering was verified and confirmed. The obtained significant benefit is that the remaining of speech component of second talker was suppressed in comparison with the previous work [26]. The degree of second speaker suppression is 10,5 (dB). This direction research can be studied and applied into other speech systems.

DMA2 is the one of the most basic MA configuration, which installed in almost acoustic equipments. Therefore, the author implemented the experiment with two opposite speaker to verify the advantage of Post-Filtering. The different phase between two noisy microphone array signals help extracting the desired talker with the DOA respectively. The proposed Post-Filtering has proven its effectiveness by saving the speaker at $\theta_1 = 0(deg)$ while suppressing the other speaker at $\theta_2 = 180(deg)$. In comparison with the

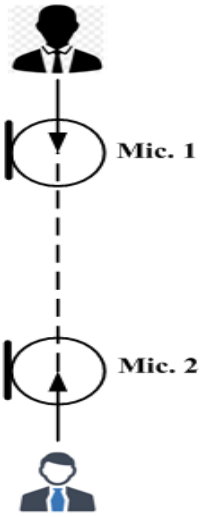



Figure 6. The acoustic recording environment in presence of two speakers.

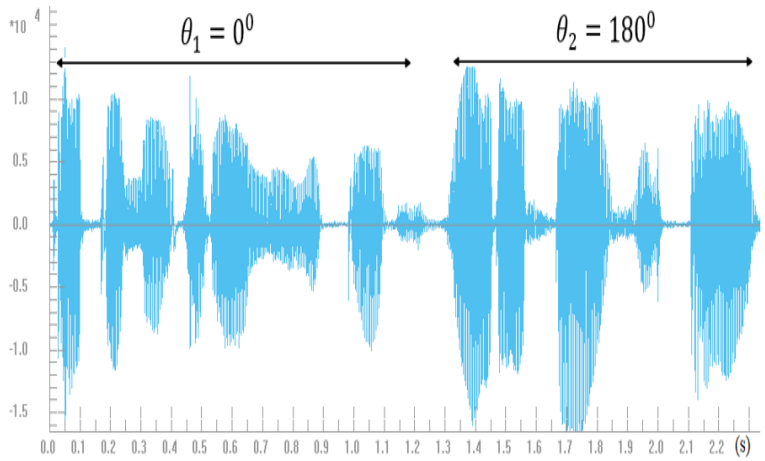



Figure 7. The waveform of microphone array signal.

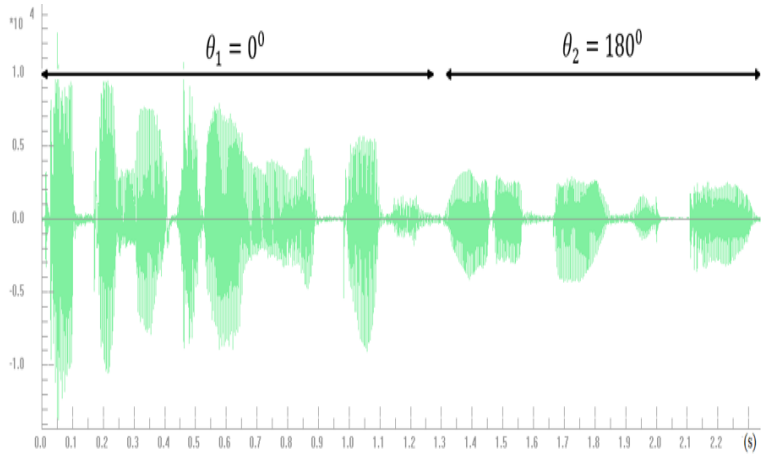



Figure 8. The processed signal by [26].

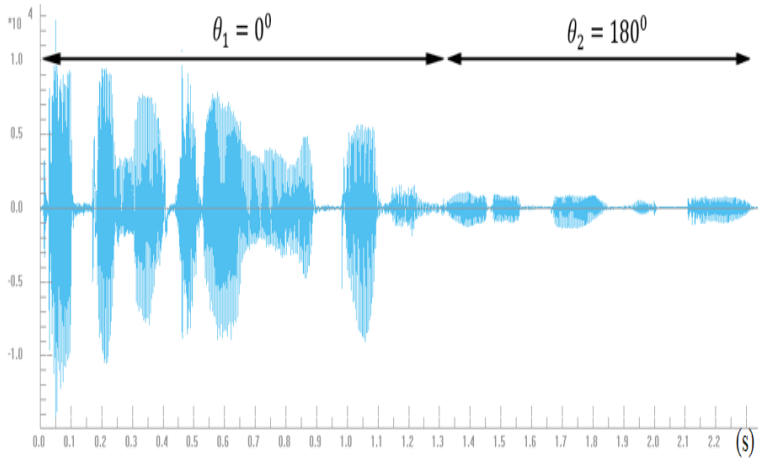



Figure 9. The obtained signal by the proposed Post-Filtering.

previous author's research, the correspondence's achievement is increasing the DIF's performance to the target speaker and removing the third-party talker in the opposite direction.

## V. CONCLUSION

A major problem for acoustic processing applications, such as hearing aids, teleconferencing, mobile devices, speech recognition is separating desired speech with good fidelity. In general, it is well-known that microphone arrays and beamforming technique for acquisition of desired target speech has significant advantages compared to single-channel approach. This paper presents a proposed post-filtering, which is suitable for extracting distant speakers while eliminating other different non-target speech. This paper describes the main stage of using DIF and the proposed post-filtering to enhance performance compared to the previous author's work, the obtained degree of suppressing second talker's speech component is to 10.5 (dB). Experimental results showed that the suggested post-filtering can be integrated into multi-microphone system for solving numerous complicated speech enhancement tasks.

## ACKNOWLEDGEMENTS

This research was supported supported by Digital Agriculture Cooperative. The author thank our colleagues from Digital Agriculture Cooperative, who provided insight and expertise that greatly assisted the research.

## REFERENCES

[1] Brandstein M., Ward D, *Microphone Arrays*, Springer-Verlag, 2001. XVIII, 398 p. doi:10.1007/978-3-66204619-7.

[2] Benesty J., Chen J. Huang Y, *Microphone Array Signal Processing*. Berlin, Germany, Springer-Verlag, 2008.240 p. doi:10.1007/978-3-54078612-2.

[3] Benesty J., Chen J., Pan C, *Fundaments of Differential Beamforming*, Springer, 2016.122 p. doi:10.1007/978-981-10-1046-0.

[4] Elko G. W., Pong A. T. N, "A steerable and variable first order differential microphone array" *Proc IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 1997, vol. 1. DOI: 10.1109/ICASSP.1997.599609.

[5] Teutsch H., Elko G. W, "First- and second- order adaptive differential microphone arrays", *Proc. IWAENC*, Sep. 2001, pp. 35–38. DOI:10.1121/1.1646972.

[6] Buck M, "Aspects of first-order differential microphone arrays in the presence of sensor imperfections", *European Transactions on Telecommunications*, vol. 13, no. 1, pp. 115–122, Mar.-Apr. 2002. DOI:10.1002/ett.4460130206.

[7] Elko G. W., Huang Y., Benesty J, *Differential microphone arrays, Audio Signal Processing for Next-Generation Multimedia Communication Systems*, MA, Kluwer Norwell, Ed., 2004.

[8] Benesty J., Chen J, "Study and Design of Differential Microphone Arrays", vol. 6, *Springer Science & Business Media*, 2013. ISBN: 978-3-642-33753-6.

[9] Elko G. W., Pong A. T. N, "A steerable and variable first order differential microphone array", *Proc IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, Apr. 1997. DOI: 10.1109/ICASSP.1997.599609.

[10] Elko G. W, Steerable and variable first-order differential microphone array, *Patent* US 006 041 127 A, Mar. 21, 2000.

[11] Elko G. W., Kubli R. A., Meyer J. M, Audio system based on at least second-order eigenbeams, *Patent* WO 2 003 061 336 A1, July 24, 2003.

[12] Elko G. W., Meyer J. M, Polyhedral audio system based on at least second-order eigenbeams, *Patent* US9 197 962 B2, Nov. 24, 2015.

[13] Sena E. D., Hacihabiboglu H., Cvetkovic Z, "On the design and implementation of higher-order differential microphones", *IEEE Trans. Audio, Speech, Lang. Process.* 20, 162–174 (2012). DOI: 10.1109/TASL.2011.2159204.

[14] Mabande E., Schad A., Kellermann K, "Design of robust superdirective beamformers as a convex optimization problem", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP-09)*, pp. 77-80, 2009-Apr. DOI: 10.1109/ICASSP.2009.4959524.

[15] Huang G., Benesty J., Cohen I., Chen J, "Differen-

tial beamforming on graphs", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. 28(1), 901–913 (2020). DOI: 10.1109/TASLP.2020.2973795.

[16] Bernardini A., Antonacci F., "Sarti A. Wave digital implementation of robust first-order differential microphone arrays", *IEEE Signal Process*. Lett. 25(2), 253–257 (2017). DOI: 10.1109/LSP.2017.2787484

[17] Borra F., Bernardini A., Antonacci F., Sarti A, "Uniform linear arrays of first-order steerable differential microphones", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. 27(12), 1906–1918 (2019). DOI: 10.1109/TASLP.2019.2934567

[18] Borra F., Bernardini A., Antonacci F., Sarti A, "Efficient implementations of first-order steerable differential microphone arrays with arbitrary planar geometry", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. (2020). DOI: 10.1109/TASLP.2020.2998283.

[19] Huang G., Chen J., Benesty J, "On the design of robust steerable frequency-invariant beampatterns with concentric circular microphone arrays", *Proc. IEEE ICASSP - Calgary*, 2018, pp. 506–510. DOI: 10.1109/ICASSP.2018.8461297

[20] Tiana-Roig E., Jacobsen F., Grande E. F,"Beamforming with a circular microphone array for localization of environmental noise sources", *J. Acoust. Soc. Am.*, 128(6), 3535–3542 (2010). https://doi.org/10.1121/1.3500669.

[21] Benesty J., Chen J, *Study and design of differential microphone arrays*, Springer-Verlag, Berlin, Germany, 2012. ISBN: 978-3-642-33753-6.

[22] Huang G., Chen J., Benesty J, "Design of planar differential microphone arrays with fractional orders", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. 28, 116–130 (2019). DOI: 10.1109/TASLP.2019.2949219.

[23] Benesty J., Cohen I., Chen J, *Array processing-Kronecker product beamforming*, Springer-Verlag, Berlin, Germany, 2019. ISBN: 978-3-030-15600-8.

[24] Cohen I., Benesty J., Chen J, "Differential Kronecker product beamforming", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. 27(5), 892–902 (2019). DOI: 10.1109/TASLP.2019.2895241.

[25] Buchris Y., Cohen I., Benesty J., Amar A, "Joint sparse concentric array design for frequency and rotationally invariant beampattern", *IEEE/ACM Trans. Audio, Speech, Lang. Process*. 28, 1143–1158 (2020). DOI: 10.1109/TASLP.2020.2982287.

[26] Stolbov M., Tatarnikova M., The Q.T. (2018), *Using Dual-Element Microphone Arrays for Automatic Keyword Recognition*, Karpov A., Jokisch O., Potapova R. (eds) Speech and Computer. SPECOM 2018. *Lecture Notes in Computer Science*, vol 11096. Springer, Cham. https://doi.org/10.1007/978-3-319-99579-3-68.

**Quản Trọng Thế** is currently working at Digital Agriculture Cooperative, Cau Giay, Hanoi, Vietnam. He received Master's Degree in Information Systems from Posts and Telecommunications Institute of Technology in 2015, and PhD in Engineering from National Research University ITMO in 2021. His research directions are information systems, signal processing, and speech enhancement.

E-mail: quantrongthe1984@gmail.com.

**Nguyễn Bá Huy** is currently a PhD student specializing in "System Analysis, Control and Information Processing" at ITMO University, Saint Petersburg, Russia. He received his Bachelor's and Master's degrees in "Control in Technical Systems" from ITMO University in 2019 and 2021 respectively. His research areas include control theory, optimization, analog and digital signal processing.

E-mail: nguyenitmo@gmail.com

**Phạm Tuấn Anh** is working as a PhD student at Russian Federation from 2017 to present, lecturer University of Information and Communication Technology Vietnamese and Korean communication - University of Danang. Field Research includes: artificial intelligence, machine vision surrounding intelligence system.

E-mail: ptanh@cit.udn.vn

# Kỹ thuật lựa chọn đặc trưng trong dự đoán code-smell dựa trên học máy

Nguyễn Thanh Bình[1], Nguyễn Hữu Nhật Minh[2], Lê Thị Mỹ Hạnh[3], Nguyễn Thanh Bình[2]
[1]Trường Cao đẳng Cơ điện-Xây dựng và Nông Lâm Trung Bộ, Bình Định
[2]Trường Đại học Công nghệ Thông tin và Truyền thông Việt-Hàn, Đại học Đà Nẵng
[3]Trường Đại học Bách khoa, Đại học Đà Nẵng

Tác giả liên hệ: Nguyễn Thanh Bình, Email: thanhbinh@cdtb.edu.vn
Ngày nhận bài: 15/03/2023, ngày sửa chữa: 20/06/2023, ngày duyệt đăng: 27/06/2023
Định danh DOI: 10.32913/mic-ict-research-vn.v2023.n1.1216

**Tóm tắt:** Các nghiên cứu thử nghiệm nhằm phát hiện code-smell trong mã nguồn bằng cách sử dụng các kỹ thuật học máy dựa trên tập dữ liệu độ đo mã nguồn đã cho thấy nhiều kết quả hứa hẹn. Các thử nghiệm đã bước đầu đưa các kỹ thuật lựa chọn đặc trưng của các tập dữ liệu vào các mô hình học máy, kết quả thử nghiệm cho thấy các kỹ thuật lựa chọn đặc trưng đã có những tác động tích cực đến hiệu suất dự đoán của các mô hình. Tuy nhiên, chưa có nghiên cứu nào so sánh hiệu suất dự đoán giữa các kỹ thuật lựa chọn đặc trưng trên cùng một mô hình trong dự đoán code-smell. Bài báo này sẽ cung cấp các kết quả thử nghiệm nhằm đánh giá toàn diện các kỹ thuật lựa chọn đặc trưng bằng cách so sánh hiệu suất dự đoán code-smell giữa các kỹ thuật này khi được áp dụng với cùng một mô hình học máy; đồng thời, bài báo này cũng cho biết kỹ thuật lựa chọn đặc trưng phù hợp nhất đối với một mô hình phát hiện code-smell dựa trên học máy.

**Từ khóa:** *Lựa chọn đặc trưng, code smell và học máy*

**Title:** **Feature selection techniques in code smell prediction based on machine learning**

**Abstract:** Recent studies that detect code smells in source code using machine learning techniques based on source code metrics datasets have shown promising results. In this work, we validate feature selection techniques of datasets with machine learning models. As a result, experiments demonstrate that using feature selection techniques could improve predictive performance of the models. To the best of our knowledge, no studies have compared the predictive performance between feature selection techniques on the same model in code smell prediction. In this paper, we would provide the experimental results to comprehensively evaluate feature selection techniques by comparing the code smell predictive performance between these techniques when being applied with the different machine learning models. Throughout extensive evaluation, it shows the best feature selection technique for machine learning-based code smell detection models.

**Keywords:** *feature selection, code smell, machine learning*

## I. GIỚI THIỆU

Trong quá trình bảo trì và phát triển, các hệ thống phần mềm được các lập trình viên cập nhật và phát triển một cách liên tục nhằm mục đích (i) thêm các cài đặt để đáp ứng các yêu cầu mới, (ii) cải thiện các chức năng hiện có, hoặc (iii) sửa các lỗi nghiêm trọng [1]. Do áp lực về thời gian, thiếu kỹ năng hoặc kinh nghiệm, các nhà phát triển không phải lúc nào cũng sẵn sàng kiểm soát hoàn toàn độ phức tạp của hệ thống để tìm ra giải pháp tối ưu trước khi áp dụng các sửa đổi [2]. Do đó, họ thường áp dụng các triển khai dưới mức tối ưu và có khả năng làm sai lệch thiết kế ban đầu của hệ thống.

Code-smell là triệu chứng của việc thiết kế hoặc triển khai dưới mức tối ưu mà các nhà phát triển áp dụng cho hệ thống phần mềm. Do đó, code-smell hiện diện trong mã nguồn có thể dẫn đến phát sinh hoặc có nguy cơ phát sinh lỗi phần mềm trong tương lai. Nhiều nghiên cứu trước đây đã chỉ ra rằng code-smell không chỉ ảnh hưởng đến việc hiểu biết về hệ thống của các nhà phát triển trong các nhiệm vụ bảo trì mà còn ảnh hưởng đến việc thiết kế của hệ thống bị thay đổi và dễ bị lỗi [3–5]. Do đó, việc phát hiện code-smell là một thách thức quan trọng đối với các nhà phát triển để giảm thiểu các nỗ lực và chi phí bảo trì hệ thống [6].

Trong những năm gần đây, các nghiên cứu thử nghiệm nhằm phát hiện code-smell trong mã nguồn bằng cách sử

dụng các kỹ thuật học máy (Machine learning techniques - MLTs) dựa trên tập dữ liệu độ đo mã nguồn (Source code metric dataset - SMD) đã cho thấy nhiều kết quả hứa hẹn [7–13]. Các thử nghiệm này cho thấy SMD là một yếu tố có vai trò quan trọng trong mô hình phát hiện code-smell dựa trên MLTs; tuy nhiên, hầu hết các nghiên cứu đều sử dụng tất cả các đặc trưng hiện hữu của tập dữ liệu (Dataset features - DFs) để huấn luyện cho các mô hình học máy. Đây không phải là một vấn đề quan trọng khi SMD chưa đủ lớn, nhưng cùng với sự phát triển nhanh chóng của các công cụ phân tích mã nguồn hiện nay, dung lượng SMD và số lượng DFs có thể thu được rất lớn, thì nó sẽ là một thách thức đối với chi phí về thời gian và tài nguyên để đáp ứng cho các mô hình phát hiện code-smell dựa trên MLTs.

Bài báo này sẽ thử nghiệm các mô hình phát hiện code-smell hiện diện trong mã nguồn dựa trên các MLTs kết hợp với các kỹ thuật lựa chọn đặc trưng (Feature selection techniques - FSTs) của SMD. FSTs là tập hợp các kỹ thuật nhằm lựa chọn từ SMD các đặc trưng có liên quan với code-smell; với sự hỗ trợ của FSTs, nhiều tập dữ liệu con (SubDataset - SD) sẽ được tạo ra dựa trên các đặc trưng được lựa chọn từ SMD. Các thử nghiệm sẽ được tiến hành bằng cách lần lượt áp dụng các SD cho các MLTs để phát hiện code-smell và trích xuất hiệu suất dự đoán (F1-score) của chúng. Các kết quả thử nghiệm này sẽ được cân nhắc để xác định FST tốt nhất đối với mỗi MLT và code-smell.

Phần tiếp theo của bài báo này là tổng quan về các nghiên cứu liên quan đến vấn đề phát hiện code-smell dựa trên các MLTs. Phần 3 của bài báo sẽ đề cập đến các cân nhắc khi lựa chọn SMD, các kỹ thuật tiền xử lý dữ liệu, các code-smell và FSTs sẽ được áp dụng cho các MLTs của các mô hình thử nghiệm. Ý tưởng thiết kế mô hình thử nghiệm dự đoán code-smell dựa trên học máy kết hợp với các kỹ thuật FSTs được trình bày trong phần 4. Phần 5 sẽ thảo luận liên quan đến kết quả thử nghiệm của các mô hình. Cuối cùng của bài báo sẽ là các kết luận chung liên quan đến các kết quả thử nghiệm và hướng nghiên cứu cần phát triển dựa trên kết quả bài báo này.

## II. CÁC NGHIÊN CỨU LIÊN QUAN

Năm 2016, Fontana và các cộng sự [7] đã sử dụng 16 MLTs để dự đoán code-smell trên SMD do nhóm phát triển; Mặc dù các kỹ thuật FSTs không được áp dụng nhưng độ chính xác dự đoán đạt đến 99,10% trong các thử nghiệm của họ. Tiếp tục với tập dữ liệu này, năm 2017 [8], nhóm nghiên cứu tiếp tục công bố kết quả phân loại mức độ nghiêm trọng của code-smell dựa trên các MLTs kết hợp với các kỹ thuật FSTs cơ bản; độ chính xác phân loại của thử nghiệm này đạt đến 93%. Mansoor và các cộng sự [9] đã thử nghiệm dự đoán code-smell trong 5 SMD mà không sử dụng đến các FSTs; độ chính xác của thử nghiệm đạt trung bình 87,00% đối với độ đo *precision* và 92,00% đối với độ đo *recall*.

Một thử nghiệm dự đoán mức độ nghiêm trọng của các lỗi của mã nguồn có sử dụng các kỹ thuật FSTs *Information Gain* và *Chi-square* để chọn các đặc trưng phù hợp từ bộ dữ liệu đã được Pushpalatha và cộng sự thực hiện [10]; độ chính xác dự đoán của mô hình thử nghiệm đạt đến 89,80%. Mohammad Y. Mhawish và các cộng sự đã tiến hành các thử nghiệm dự đoán code-smell dựa trên các mô hình học máy có sử dụng các kỹ thuật FSTs [11, 12] trên tập dữ liệu của Fontana [7]; độ chính xác dự đoán của các thử nghiệm được cải thiện dần từ 98,38% đến 99,70% tùy thuộc vào các kỹ thuật tiền xử lý dữ liệu.

Gần đây, nhóm nghiên cứu của Dewangan đã tiến hành xây dựng mô hình dự đoán code-smell dựa trên 6 MLTs [13] để dự đoán code-smell *Long-method* trên tập dữ liệu của Fontana [7]; họ đã áp dụng các kỹ thuật FSTs *Chi-square* và *Wrapper based* để lựa chọn các đặc trưng phù hợp với mô hình thử nghiệm, đồng thời phương pháp xác thực *10-Fold- Cross-Validation* được áp dụng để xác thực độ tin cậy của việc dự đoán code-smell của mô hình thử nghiệm. Kết quả thử nghiệm cho thấy dự đoán code-smell *Feature-envy* đạt độ chính xác đến 99,12%.

Các nghiên cứu trên đã bước đầu đưa các kỹ thuật FSTs của tập dữ liệu vào các mô hình để tăng hiệu suất dự đoán. Kết quả thử nghiệm cho thấy các kỹ thuật FSTs đã có những tác động tích cực đến hiệu suất dự đoán của các mô hình. Tuy nhiên, chưa có nghiên cứu nào so sánh khả năng tăng hiệu suất giữa các kỹ thuật FSTs đối với cùng một mô hình dự đoán code-smell. Vì vậy, bài báo này sẽ bổ sung cho các các nghiên cứu trên trong hai vấn đề chính:

(i) Cung cấp một nghiên cứu nhằm đánh giá toàn diện các FSTs bằng cách so sánh hiệu suất dự đoán code-smell giữa các FSTs khi được áp dụng với cùng một mô hình thử nghiệm.

(ii) Đồng thời, bài báo này cũng cho biết kỹ thuật FST tốt nhất đối với một mô hình phát hiện code-smell dựa trên MLTs.

## III. LỰA CHỌN CÁC FSTS

Việc trích xuất độ đo từ mã nguồn thường tạo ra các SMD có số lượng các đặc trưng khá lớn tùy thuộc vào công cụ được sử dụng; Mặc dù vậy, không phải lúc nào các các đặc trưng hiện diện trong SMD đều liên quan trực tiếp đến các code-smell cần được phát hiện, cho nên các MLTs có thể tạo ra kết quả kém chính xác và khó hiểu hơn hoặc có thể không phát hiện ra bất kỳ điều gì hữu ích.

FSTs là các phương pháp cố gắng xác định và loại bỏ càng nhiều các đặc trưng không liên quan và dư thừa càng tốt trước khi áp dụng vào các MLTs. FSTs có thể nâng cao

hiệu suất, tiết kiệm thời gian nhờ việc giảm không gian tìm kiếm trên các biến độc lập và trong một số trường hợp, giảm yêu cầu lưu trữ. Hiện nay, các phương pháp lựa chọn đặc trưng được chia làm ba nhóm sau: phương pháp Filter, phương pháp Wrapper và phương pháp Embedded.

### 1. Phương pháp Filter (Filter methods)

*a) Pearson correlation (F-Corr):*

Pearson correlation là một hệ số có giá trị từ -1 đến 1; giá trị này cho biết mức độ mà hai đặc trưng có liên quan tuyến tính với nhau. Nếu Pearson correlation = 0, nghĩa là chúng không có liên quan với nhau. Mục đích của kỹ thuật này là xác định các đặc trưng có hệ số Pearson correlation cao và loại bỏ chúng khỏi tập đặc trưng của SMD.

*b) Analysis of Variance (F-Aov):*

là một kỹ thuật thống kê kiểm tra xem các đặc trưng đầu vào khác nhau có giá trị khác nhau đáng kể cho biến đầu ra hay không. Kỹ thuật cho phép phân tích nhiều nhóm dữ liệu để xác định sự khác biệt giữa các mẫu và trong các mẫu, để có được thông tin về mối quan hệ giữa các biến phụ thuộc và biến độc lập giúp chúng ta lựa chọn các đặc trưng tốt nhất.

*c) Mutual information (F-MI):*

MI là lượng thông tin có thể thu được từ một biến ngẫu nhiên cho một biến khác. Giá trị này thể hiện mức độ phụ thuộc lẫn nhau giữa các biến độc lập và biến phụ thuộc. MI luôn lớn hơn hoặc bằng 0, trong đó giá trị càng lớn thì mối quan hệ giữa hai biến càng lớn. Nếu MI= 0, thì các biến là độc lập.

### 2. Phương pháp Wrapper (Wrapper methods)

*a) Forward selection (W-Fws):*

Một đặc trưng tốt nhất được chọn để bắt đầu và thêm nhiều lần lặp lại. Ở mỗi lần lặp lại tiếp theo, các đặc trưng tốt nhất còn lại được thêm vào dựa trên tiêu chí hiệu suất của chúng.

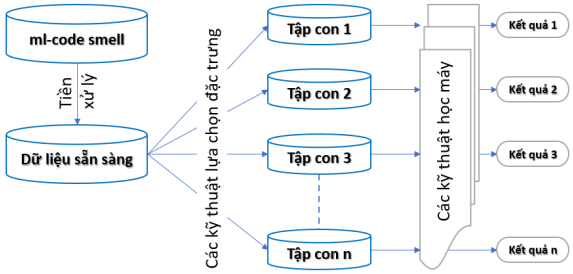
*b) Backward elimination (W-Bwe):*

Kỹ thuật này bắt đầu với tất cả các đặc trưng và lặp đi lặp lại loại bỏ từng đặc trưng một. Một trong những thuật toán phổ biến nhất là Recursive Feature Elimination (RFE) giúp loại bỏ các đặc trưng ít quan trọng hơn dựa trên xếp hạng mức độ quan trọng của chúng.

### 3. Phương pháp Embedded (Embedded methods)

Các phương pháp thuộc nhóm Embedded kết hợp các ưu điểm của cả hai phương pháp Filter và phương pháp Wrapper. Các phương pháp này hiệu quả hơn các phương pháp Wrapper và chính xác hơn các phương pháp Filter; trong các phương pháp Embedded, việc lựa chọn đặc trưng được thực hiện trong quá trình huấn luyện. Hai phương pháp lựa chọn đặc trưng thuộc nhóm Embedded là ***Hồi quy Ridge*** **(E-Ridge)** và ***Hồi quy Lasso*** **(E-Lasso)** được lựa chọn để đưa vào các mô hình thử nghiệm.

## IV. THIẾT KẾ VÀ THỬ NGHIỆM MÔ HÌNH DỰ ĐOÁN CODE-SMELL KẾT HỢP FSTS




Hình 1. Mô hình thử nghiệm xác định hiệu suất của các kỹ thuật lựa chọn đặc trưng

### 1. Lựa chọn SCM và kỹ thuật tiền xử lý dữ liệu

Bài báo sử dụng tập dữ liệu **ml-Codesmell** của nhóm tác giả và đã được công bố [14]. Tập dữ liệu này đã được đề xuất cho các mô hình dự đoán code-smell dựa trên phương pháp học máy với một lượng lớn quan sát cùng với nhiều đặc trưng được tạo bằng cách thu thập các độ đo mã nguồn được trích xuất từ 525 dự án nguồn mở.

Tập dữ liệu ml-Code-smell được phân tách thành 2 tập dữ liệu con: tập dữ liệu huấn luyện *(training)* để huấn luyện các MLTs và tập dữ liệu kiểm tra *(testing)* để kiểm tra hiệu suất dự đoán của các MLTs đối với sự hiện diện của các code-smell có trong tập này. Dựa trên kết quả đã thu được qua thử nghiệm các tỷ lệ phân tách *training:testing* khác nhau trên cùng 1 tập dữ liệu và kỹ thuật học máy [14], tỷ lệ phân tách dữ liệu *training:testing* là *80:20* được chọn để áp dụng cho các thử nghiệm của bài báo này.

Kỹ thuật cân bằng dữ liệu SMOTE được áp dụng để xử lý mất cân bằng dữ liệu trong tập dữ liệu huấn luyện. SMOTE là một trong các kỹ thuật lấy mẫu quá mức *(Oversampling)*; trong đó các mẫu tổng hợp được tạo ra cho lớp thiểu số. Thuật toán này giúp khắc phục vấn đề *over-fitting* do quá trình lấy mẫu ngẫu nhiên *(Random Oversampling)* gây ra. Nó tập trung vào không gian các đặc trưng để tạo ra các mẫu quan sát mới với sự trợ giúp của phép nội suy giữa lớp thiểu số nằm cùng nhau.

## 2. Lựa chọn các code-smell cho mô hình thử nghiệm

Tập dữ liệu ml-Codesmell [1], bao gồm 8 code-smell ở cấp độ lớp *(Class-level)* và 6 code-smell ở cấp độ phương thức *(Method-level)*, được thể hiện trong Bảng I.

Bảng I
TÓM LƯỢC THÔNG TIN VỀ TẬP DỮ LIỆU ML-CODESMELL

| Cấp độ code-smell | Dung lượng | Số lượng mẫu | Số lượng đặc trưng | Số lượng code-smell |
|---|---|---|---|---|
| Class-level | 76 MB | 373,400 | 41 | 8 |
| Method-level | 491 MB | 2,486,282 | 33 | 6 |

Trong phạm vi bài báo, để tập trung vào mục tiêu đánh giá hiệu suất của các FSTs và tiết kiệm thời gian, năm code-smell ở cấp độ lớp *(Brain-Class, Data-Class, God-Class, Model-Class, và Schizophrenic-Class)* được lựa chọn để đưa vào mô hình thử nghiệm.

## 3. Các mô hình học máy (MTLs)

Các mô hình học máy được tạo ra với nhiều thuật toán khác nhau. Năm thuật toán được chọn dựa trên mức độ phổ biến, tính đơn giản và hiệu suất cao của chúng. Các thuật toán này được mô tả ngắn gọn như sau:

- **Random Forest Classifier (RFC)**: RFC phù hợp với một số bộ phân loại cây quyết định dựa vào các tập con khác nhau của SMD và tập hợp các cây đầu ra theo giá trị trung bình của từng cây. Độ chính xác đầu ra của RFC phụ thuộc vào sức mạnh của từng cây trong rừng và mối tương quan giữa chúng. Do đó, RFC có thể cải thiện độ chính xác dự đoán và tránh vấn đề khớp quá mức (over-fitting) [15].
- **Light Gradient Boosting (LGB)**: LGB xây dựng một mô hình dựa trên các cây quyết định đơn giản không được tối ưu hóa tốt và sau đó khái quát hóa chúng bằng cách tối ưu hóa một hàm mất mát (loss-function) được xác định tùy ý để đưa ra các dự đoán mạnh mẽ hơn. Các ưu điểm của LGB như sau: (i) có tốc độ huấn luyện nhanh hơn và hiệu quả cao hơn nhiều thuật toán khác nhưng tiêu tốn bộ nhớ thấp hơn, (ii) có độ chính xác cao hơn bất kỳ thuật toán tăng tốc nào khác và (iii) có khả năng tương thích tốt hơn với các bộ dữ liệu lớn [16].
- **K-nearest Neighbors Classifier (KNN)**: Các quan sát được phân loại dựa trên lớp của mẫu lân cận gần nhất của chúng. KNN có hai giai đoạn: (i) Xác định các mẫu gần nhất và (ii) xác định lớp sử dụng các lân cận đó. Một số ưu điểm của KNN như sau: (i) nhanh chóng triển khai và gỡ lỗi, (ii) rất hiệu quả nếu các lân cận được sử dụng phân tích để làm giải thích và (iii) có nhiều kỹ thuật có thể nhanh chóng tăng độ chính xác và cải thiện đáng kể thời gian quá trình thử nghiệm trên các tập dữ liệu lớn [17].
- **Linear Logistic Regression (LLR)**: Thuật toán mạnh mẽ này cho phép nhiều biến độc lập được phân tích đồng thời và giảm các tác động gây nhiễu bằng cách phân tích mối liên hệ của tất cả các biến. LLR phân loại các mẫu dựa trên xác suất của một sự kiện, vì vậy các biến phụ thuộc và độc lập của hồi quy logistic không cần mối tương quan [18, 19].
- **Linear Support Vector Classification (SVM)**: SVM là một mô hình học máy có giám sát (supervised learning model) với sự kết hợp các thuật toán học máy nhằm phân tích dữ liệu để phân loại và phân tích hồi quy. SVM hoạt động tương đối tốt khi có biên độ phân cách rõ ràng giữa các lớp. SVM hiệu quả hơn trong không gian nhiều chiều và tương đối hiệu quả về bộ nhớ [20].

## 4. Lựa chọn các độ đo hiệu suất để đánh giá mô hình

Sau khi được huấn luyện với *training*, các MLTs có thể dự đoán một mẫu trong *testing* là *dương tính*(là code-smell) hay là *âm tính* (không phải code-smell). Kết quả dự đoán của mô hình có thể rơi vào một trong 4 trường hợp sau:

- TP (True-Positive): Dự đoán đúng 1 mẫu *dương tính*.
- TN (True-Negative): Dự đoán đúng 1 mẫu *âm tính*.
- FP (False-Positive): Dự đoán sai 1 mẫu *dương tính*.
- FN (False-Negative): Dự đoán sai 1 mẫu *âm tính*.

Sau khi xác định được các giá trị trên, độ chính xác dự đoán của mô hình thử nghiệm được đánh giá bằng các độ đo cơ bản sau:

- **Accuracy** *Accuracy* là một độ đo dùng để đánh giá hiệu suất dự đoán của một MLTs. Giá trị *Accuracy* càng cao thì độ chính xác dự đoán càng lớn. *Accuracy* được tính toán dựa trên công thức sau:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Tuy nhiên, khi tập dữ liệu bị mất cân bằng nghiêm trọng, việc sử dụng *Accuracy* làm độ đo đánh giá hiệu suất của mô hình thường không hiệu quả vì hầu hết chúng đều có độ chính xác rất cao; bất kỳ mô hình nào có thể dự đoán tất cả các mẫu thuộc về lớp đa số cũng sẽ cho kết quả gần 100%. Do đó, các độ đo như *Precision*, *Recall* và *F1-score* thường được cân nhắc để thay thế cho *Accuracy*. Các độ đo này đề cao tính chính xác của lớp thiểu số hơn là lớp đa số.

[1] https://doi.org/10.6084/m9.figshare.21343299.v2

- **Precision** Độ đo này nhằm cố gắng trả lời câu hỏi sau: Tỷ lệ nhận dạng đúng các mẫu *dương tính* là bao nhiêu? *Precision* được định nghĩa như sau:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- **Recall** *Recall* định lượng tỷ lệ dự đoán đúng mẫu *dương tính* từ tất cả các dự đoán *dương tính* đã được thực hiện. Không giống như *Precision* chỉ trả về các dự đoán chính xác mẫu *dương tính* trong số tất cả các dự đoán *dương tính*, *Recall* còn cung cấp dấu hiệu của các dự đoán *dương tính* bị bỏ lỡ. Công thức tính *Recall* như sau:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- **F1-score** Đây là một độ đo hài hòa giữa *Precision* và *Recall*, đồng thời cũng là độ đo lý tưởng để đánh giá các mô hình có bộ dữ liệu mất cân bằng nghiêm trọng.

$$F1\text{-}score = \frac{2(Precision * Recall)}{Precision + Recall} \quad (4)$$

Trong bài báo này, *F1-score* được sử dụng làm thước đo hiệu suất của các mô hình dự đoán dựa trên MLTs.

## V. KẾT QUẢ VÀ THẢO LUẬN

### 1. Kết quả dự đoán đối với các code-smell

*a) Brain-Class:*

Kết quả dự đoán code-smell Brain-Class được thể hiện trong Bảng II. Hầu hết các FSTs đều cho hiệu suất tốt hơn so với SMD nguyên bản, ngoại trừ đối với mô hình MLTs áp dụng giải thuật LGB; Nhìn chung, hiệu suất dự đoán của các MLTs đối với code-smell này không cao; tuy nhiên, khi LGB sử dụng FSTs là F-Corr thì hiệu suất đạt đến 95%, tương đương với SMD nguyên bản.

Bảng II
KẾT QUẢ HIỆU SUẤT DỰ ĐOÁN CODE-SMELL BRAIN-CLASS

| FSTs \ MLTs | KNN | LGB | LRR | RFC | SVM |
|---|---|---|---|---|---|
| None-FST | 0,30 | 0,95 | 0,15 | 0,47 | 0,43 |
| E-Lasso | 0,31 | 0,66 | 0,12 | 0,48 | 0,46 |
| E-Ridge | 0,26 | 0,64 | **0,32** | 0,5 | 0,42 |
| F-Aov | 0,33 | 0,58 | 0,30 | 0,50 | 0,48 |
| F-Corr | 0,33 | **0,95** | 0,17 | **0,5** | 0,45 |
| F-MI | 0,28 | 0,63 | 0,24 | 0,48 | **0,51** |
| W-Bwe | **0,5** | 0,68 | 0,24 | 0,39 | 0,5 |
| W-Fws | 0,42 | 0,55 | 0,21 | 0,40 | 0,20 |

*b) Data-Class:*

Số liệu của kết quả dự đoán được thể hiện trong Bảng III. Hiệu suất dự đoán của các MLTs đối với code-smell Data-Class vượt trội và ổn định so với Brain-Class; Hầu hết các FSTs đều có hiệu suất dự đoán đạt từ 99% trở lên tùy theo MLTs được chọn.

Bảng III
KẾT QUẢ HIỆU SUẤT DỰ ĐOÁN CODE-SMELL DATA-CLASS

| FSTs \ MLTs | KNN | LGB | LRR | RFC | SVM |
|---|---|---|---|---|---|
| None-FST | 0,85 | 1 | 0,95 | 1 | 1 |
| E-Lasso | 0,86 | **1** | 0,9 | **1** | **1** |
| E-Ridge | 0,96 | 0,99 | 0,92 | 0,97 | 0,95 |
| F-Aov | 0,98 | 1 | 0,95 | 0,99 | 0,96 |
| F-Corr | 0,89 | 1 | **0,96** | 0,99 | 0,99 |
| F-MI | 0,99 | 1 | 0,95 | 0,99 | 0,95 |
| W-Bwe | **0,99** | 1 | 0,95 | 0,96 | 0,99 |
| W-Fws | 0,98 | 1 | 0,95 | 0,99 | 1 |

*c) God-Class:*

Kết quả hiệu suất thử nghiệm với code-smell God-Class, được trình bày trong Bảng IV, cho thấy chỉ có 2 MLTs là LGB và RFC là phù hợp với code-smell này khi cho thấy hiệu suất dự đoán đạt từ 99%; tuy nhiên, các kỹ thuật FSTs không phát huy hiệu quả khi hiệu suất của các mô hình không tỏ ra vượt trội so với SMD nguyên bản.

Bảng IV
KẾT QUẢ HIỆU SUẤT DỰ ĐOÁN CODE-SMELL GOD-CLASS

| FSTs \ MLTs | KNN | LGB | LRR | RFC | SVM |
|---|---|---|---|---|---|
| None-FST | 0,62 | **1** | 0,36 | 0,97 | 0,53 |
| E-Lasso | 0,64 | 0,96 | 0,24 | 0,95 | 0,48 |
| E-Ridge | 0,6 | 0,96 | 0,5 | 0,95 | 0,47 |
| F-Aov | 0,62 | 0,96 | **0,54** | 0,95 | 0,3 |
| F-Corr | 0,56 | 0,97 | 0,24 | 0,87 | 0,59 |
| F-MI | **0,71** | 1 | 0,44 | **0,99** | 0,58 |
| W-Bwe | 0,65 | 0,87 | 0,52 | 0,66 | **0,64** |
| W-Fws | 0,65 | 0,87 | 0,52 | 0,66 | 0,64 |

*d) Model-Class:*

Hầu hết các MLTs được đưa vào mô hình tỏ ra phù hợp với code-smell Model-Class bởi vì chúng đều trả về hiệu suất rất cao (từ 98% trở lên) khi thử nghiệm, ngoại trừ kỹ thuật SVM; tuy nhiên, bên cạnh đó, các phương pháp FSTs lại không có hiệu quả như kỳ vọng khi hiệu suất của chúng chênh lệch không đáng kể so với SMD nguyên bản mặc dù tiêu tốn khá nhiều thời gian cho quá trình loại bỏ bớt các DFs không liên quan và dư thừa (kết quả thử nghiệm có trong Bảng V).

Bảng V
KẾT QUẢ HIỆU SUẤT DỰ ĐOÁN CODE-SMELL MODEL-CLASS

| FSTs \ MLTs | KNN | LGB | LRR | RFC | SVM |
|---|---|---|---|---|---|
| None-FST | 1 | 1 | 0,98 | 1 | 0,77 |
| E-Lasso | 1 | 1 | **0,98** | 1 | 0,64 |
| E-Ridge | 1 | 1 | 0,98 | 1 | **1** |
| F-Aov | 0,98 | 0,98 | 0,96 | 0,97 | 0,92 |
| F-Corr | 1 | **1** | 0,98 | **1** | 0,24 |
| F-MI | **1** | 1 | 0,98 | 1 | 0,31 |
| W-Bwe | 0,97 | 0,98 | 0,94 | 0,98 | 0,91 |
| W-Fws | 0,97 | 0,98 | 0,94 | 0,98 | 0,91 |

*e) Schizophrenic-Class*

Khi thử nghiệm dự đoán code-smell Schizophrenic-Class bằng các mô hình áp dụng các MLTs LGB và RFC, các FSTs không cho thấy sự chênh lệch hiệu suất dự đoán so với SMD nguyên bản bởi vì tất cả chúng đều trả về hiệu suất 100% (Bảng VI); tuy nhiên, khi dùng các mô hình được áp dụng các MLTs còn lại, các FSTs đã thể hiện được tính hiệu quả của chúng: tăng hiệu suất từ 58% lên 86% đối với MLTs KNN, tăng từ 84% lên 99% đối với MLTs LRR và tăng từ 40% lên 53% đối với MLTs SVM.

Bảng VI
KẾT QUẢ HIỆU SUẤT DỰ ĐOÁN CODE-SMELL SCHIZOPHRENIC-CLASS

| FSTs \ MLTs | KNN | LGB | LRR | RFC | SVM |
|---|---|---|---|---|---|
| None-FST | 0,58 | 1 | 0,84 | 1 | 0,4 |
| E-Lasso | 0,64 | **1** | 0,69 | 1 | **0,53** |
| E-Ridge | 0,73 | 1 | 0,88 | 1 | 0,34 |
| F-Aov | 0,66 | 1 | 0,98 | 1 | 0,4 |
| F-Corr | 0,63 | 1 | 0,84 | **1** | 0,35 |
| F-MI | 0,75 | 1 | **0,99** | 1 | 0,4 |
| W-Bwe | **0,86** | 1 | 0,99 | 1 | 0,34 |
| W-Fws | 0,79 | 1 | 0,99 | 1 | 0,4 |

## 2. Thảo luận chung về các kết quả thử nghiệm

Kết quả thử nghiệm cho thấy sau khi tập dữ liệu được áp dụng các kỹ thuật FSTs, hiệu suất dự đoán của các mô hình MLTs đã có những cải thiện tích cực tùy theo từng FSTs và MLTs. Dựa vào các kết quả thử nghiệm trên, các FSTs đã thể hiện một số đặc điểm tích cực sau:

- Cải thiện độ chính xác của mô hình: Bằng cách chọn kỹ thuật FSTs phù hợp với MLTs, việc lựa chọn đặc trưng đã thể hiện khả năng cải thiện độ chính xác của mô hình thử nghiệm. Tất cả các thử nghiệm đều cho thấy với mỗi MLTs đều có ít nhất một kỹ thuật FSTs cho hiệu suất bằng hoặc cao hơn với tập dữ liệu nguyên bản SMD.

- Cải thiện khả năng diễn giải: Bằng cách chỉ chọn các đặc trưng phù hợp nhất, các kỹ thuật FSTs có thể cải thiện khả năng diễn giải của mô hình, giúp dễ hiểu và giải thích các đặc trưng góp phần vào dự đoán của mô hình.

- Giảm dung lượng tập dữ liệu: Các FSTs giúp giảm kích thước của tập dữ liệu, điều này có thể giúp trực quan hóa và phân tích dữ liệu dễ dàng hơn.

- Giảm thời gian huấn luyện các mô hình: Bằng cách giảm số lượng đặc trưng, FSTs có thể giảm độ phức tạp tính toán của mô hình, vì vậy thời gian huấn luyện các mô hình được giảm đi đáng kể.

Nhìn chung, kết quả thử nghiệm đã cho thấy các FSTs giúp giảm thiểu thời gian huấn luyện, cải thiện độ chính xác, hiệu quả và khả năng diễn giải của các mô hình học máy, giúp chúng trở nên hiệu quả và hữu ích hơn cho nhiều ứng dụng. Tuy nhiên, quá trình thử nghiệm cũng cho thấy một số vấn đề cần cân nhắc khi áp dụng các FSTs:

- Tốn thời gian: FSTs có thể là một quá trình tốn nhiều thời gian, đặc biệt khi làm việc với các bộ dữ liệu lớn hoặc sử dụng các kỹ thuật thuộc nhóm phương pháp Wrapper.

- Mất thông tin: Việc xóa các đặc trưng khỏi tập dữ liệu có thể dẫn đến mất thông tin có giá trị tiềm ẩn, điều này có thể ảnh hưởng đến độ chính xác của mô hình. Kết quả thử nghiệm cho thấy nhiều FSTs đã cho hiệu suất dự đoán thấp hơn so với tập dữ liệu nguyên bản (None-FSTs).

Vì vậy, khi áp dụng các FSTs, điều quan trọng là phải đánh giá cẩn thận các ưu điểm và nhược điểm tiềm ẩn của từng kỹ thuật FSTs và chọn kỹ thuật tốt nhất dựa trên các mục tiêu và yêu cầu cụ thể của mô hình học máy.

## VI. KẾT LUẬN

Đến phần cuối bài báo này, qua kết quả thử nghiệm, có thể kết luận rằng FSTs là các kỹ thuật về việc giữ các đặc trưng hữu ích và có liên quan một cách tự động thông qua các quy trình đã được áp dụng trong các mô hình thử nghiệm. Bài báo đã thử nghiệm các kỹ thuật FSTs được xây dựng từ các quan điểm khác nhau: từ số liệu thống kê đến phương pháp học máy. Nhìn chung, khi áp dụng FSTs cho các MLTs, có thể dẫn đến các mô hình tốt hơn, đạt được hiệu suất cao hơn và dễ hiểu hơn. Cuối cùng nhưng không kém phần quan trọng, đó là việc có một tập hợp con các đặc trưng cho phép mô hình học máy được huấn luyện nhanh hơn mà vẫn cải thiện được hiệu suất dự đoán. Về hướng nghiên cứu sâu hơn đối với các kỹ thuật FSTs, việc kết hợp nhiều kỹ thuật FSTs để lựa chọn các đặc trưng cần được tiếp cận; một tập hợp các kỹ thuật FSTs phù hợp có thể cải thiện được hiệu suất và giảm thiểu thời gian huấn luyện cho các mô hình học máy trên tập dữ liệu siêu lớn.

## TÀI LIỆU THAM KHẢO

[1] M. Lehman, “Programs, life cycles, and laws of software evolution,” *Proceedings of the IEEE*, vol. 68, pp. 1060–1076, 1980.

[2] D. Parnas, “Software aging.” IEEE Comput. Soc. Press, 1994, pp. 279–287.

[3] P. Avgeriou and P. Kruchten, “Managing technical debt in software engineering,” *Dagstuhl Seminar 16162*, 2016.

[4] F. Khomh, M. D. Penta, Y.-G. Guéhéneuc, and G. Antoniol, “An exploratory study of the impact of antipatterns on class change- and fault-proneness,” *Empirical Software Engineering*, vol. 17, pp. 243–275, 8 2012.

[5] D. Spadini, F. Palomba, A. Zaidman, M. Bruntink, and A. Bacchelli, “On the relation of test smells to software code quality,” *2018 IEEE International*

*Conference on Software Maintenance and Evolution (ICSME)*, pp. 1–12, 8 2018.

[6] A. Yamashita and L. Moonen, "Do code smells reflect important maintainability aspects?" *IEEE International Conference on Software Maintenance, ICSM*, pp. 306–315, 8 2012.

[7] F. A. Fontana, M. V. Mäntylä, M. Zanoni, and A. Marino, "Comparing and experimenting machine learning techniques for code smell detection," *Empirical Software Engineering*, vol. 21, pp. 1143–1191, 6 2016.

[8] F. A. Fontana and M. Zanoni, "Code smell severity classification using machine learning techniques," *Knowledge-Based Systems*, vol. 128, pp. 43–58, 7 2017.

[9] U. Mansoor, M. Kessentini, B. R. Maxim, and K. Deb, "Multi-objective code-smells detection using good and bad design examples," *Software Quality Journal*, vol. 25, pp. 529–552, 6 2017.

[10] P. M. N and M. M, "Predicting the severity of open source bug reports using unsupervised and supervised techniques," *International Journal of Open Source Software and Processes*, vol. 10, pp. 1–15, 1 2019.

[11] M. Y. Mhawish and M. Gupta, "Generating code-smell prediction rules using decision tree algorithm and software metrics," *International Journal of Computer Sciences and Engineering*, vol. 7, pp. 41–48, 5 2019.

[12] ——, "Predicting code smells and analysis of predictions: Using machine learning techniques and software metrics," *Journal of Computer Science and Technology*, vol. 35, pp. 1428–1445, 11 2020.

[13] S. Dewangan and R. S. Rao, *Code Smell Detection Using Classification Approaches*, 2022.

[14] B. N. Thanh, M. N. N. H, H. L. T. My, and B. N. Thanh, "Ml-codesmell: A code smell prediction dataset for machine learning approaches." Association for Computing Machinery, 12 2022, pp. 368–374.

[15] L. Breiman, "Random forests," pp. 5–32, 2001.

[16] D. A. McCarty, H. W. Kim, and H. K. Lee, "Evaluation of light gradient boosted machine learning technique in large scale land use and land cover classification," *Environments*, vol. 7, p. 84, 10 2020.

[17] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers - a tutorial," *ACM Computing Surveys*, vol. 54, pp. 1–25, 7 2022.

[18] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, pp. 12–18, 2014.

[19] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The Journal of Educational Research*, vol. 96, pp. 3–14, 9 2002.

[20] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," pp. 1–14, 2001.

**Nguyễn Thanh Bình** tốt nghiệp Thạc sỹ chuyên ngành Khoa học máy tính năm 2011 và đang làm Nghiên cứu sinh chuyên ngành Khoa học máy tính tại Đại học Đà Nẵng. Hiện công tác tại Phòng Đào tạo và Hợp tác quốc tế, Trường Cao đẳng Cơ điện-Xây dựng và Nông Lâm Trung Bộ, Bình Định.
Email: thanhbinh@cdtb.edu.vn

**Nguyễn Hữu Nhật Minh** tốt nghiệp Đại học chuyên ngành Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa TP. Hồ Chí Minh năm 2013. Năm 2020 bảo vệ Tiến sĩ ngành Khoa học và Kỹ thuật Máy tính của trường Đại học Kyung Hee, Hàn Quốc. Hiện là giảng viên, nghiên cứu viên của trường Đại học Công nghệ Thông tin & Truyền thông Việt Hàn, Đại học Đà Nẵng., phụ trách Chương trình nghiên cứu Viện Khoa học và Công nghệ Số của trường. Các lĩnh vực nghiên cứu chính bao gồm wireless communications, mobile edge computing, federated learning, xử lý ngôn ngữ tự nhiên và xử lý hình ảnh.
Email: nhnminh@vku.udn.vn

**Lê Thị Mỹ Hạnh** hiện là giảng viên Khoa Công nghệ Thông tin, Trường Đại học Bách Khoa Đà Nẵng, Việt Nam. Nhận bằng Thạc sỹ năm 2004 và bằng Tiến sĩ Khoa học Máy tính năm 2016, tại Đại học Đà Nẵng. Các lĩnh vực nghiên cứu gồm Công nghệ phần mềm và Ứng dụng các kỹ thuật heuristic cho các vấn đề trong công nghệ phần mềm.
Email: ltmhanh@dut.udn.vn

**Nguyễn Thanh Bình** nhận bằng Tiến sĩ Công nghệ Thông tin tại Trường Đại học Bách Khoa Grenoble, Cộng hòa Pháp năm 2004 và đã được công nhận Phó Giáo sư từ năm 2013. Hiện công tác tại Trường Đại học Công nghệ Thông tin và Truyền thông Việt-Hàn, Đại học Đà Nẵng. Các lĩnh vực nghiên cứu chính bao gồm Công nghệ phần mềm và Chất lượng phần mềm.
Email: ntbinh@vku.udn.vn

# Using LSTM Network for Predicting Radio Mobile Networks' Behaviors

Cuong Pham-Quoc[1], Tuan Tang-Anh[2], Chien Tang-Tan[3]

[1] MobiFone Corporation, Hanoi, Vietnam

[2] Faculty of Electronics and Telecommunication Engineering, University of Science and Technology, The University of Danang, Danang, Vietnam

[3] Faculty of Computer and Electronics Engineering, Vietnam – Korea University of Information and Communication Technology, The University of Danang, Danang, Vietnam

Correspondence: Cuong Pham-Quoc, cuong.phamquoc@mobifone.vn

Received: 28/02/2023, revised: 23/06/2023, accepted: 30/6/2023

Digital Object Identifier:: 0.32913/mic-ict-research-vn.v2023.n1.1212

**Abstract:** Breakthroughs in artificial intelligence, machine learning and deep learning with great recent achievements have been strongly impacting the mobile communication industry. The traditional method of mobile network management and operation has been shifting toward automation and proactive monitoring. Forecasting radio mobile networks' behaviors in coexisting multi-layer technology mobile networks for improving resource allocation efficiency and optimizing network's key performance indicators becomes a critical task. The use of a Long-Short Term Memory (LSTM) network for predicting radio networks' behaviors based on real statistical data from operational support systems is suggested in this research. The promising results show the potential of our approach. The prediction results can help mobile network operators leverage networks' operating performance.

**Keywords:** *Long-Short Term Memory, predicting, behavior, radio mobile network, data.*

**Tiêu đề:** **Sử dụng mạng LSTM để dự báo hành vi mạng thông tin vô tuyến di động**

**Tóm tắt:** Những đột phá trong lĩnh vực trí tuệ nhân tạo, học máy và học sâu với các thành tựu vượt trội đã tác động mạnh mẽ đến ngành công nghiệp thông tin di động. Các phương pháp vận hành khai thác, quản lý mạng lưới truyền thống đang từng bước dịch chuyển sang hướng tự động hóa và chủ động giám sát. Việc dự báo các hành vi mạng thông tin vô tuyến di động trong mạng lưới thông tin tích hợp đa công nghệ nhằm cải thiện hiệu quả cấp phát tài nguyên và tối ưu các chỉ số chất lượng mạng lưới là nhiệm vụ cấp thiết. Nghiên cứu này đề xuất sử dụng mạng Long-Short Term Memory (LSTM) để dự báo hành vi mạng lưới dựa trên tập dữ liệu thống kê thực tế từ hệ thống hỗ trợ vận hành khai thác OSS (Operational Support System). Các kết quả dự báo cho thấy tiềm năng của phương pháp mà bài báo đề xuất, Với những kết quả đạt được có thể giúp các nhà mạng di động tối ưu hiệu năng vận hành khai thác mạng lưới.

**Từ khóa:** *Long-Short Term Memory, dự báo, hành vi, mạng thông tin vô tuyến di động, dữ liệu.*

## I. INTRODUCTION

Nowadays, with a tremendous volume of data and advances in computing power, deep learning (DL) has played a key role in complicated challenges. It is anticipated that next-generation mobile networks would combine DL and advanced machine learning (ML) to complete tasks that previously seemed impossible [1]. The world's leading telecommunication vendors, such as Nokia and Ericsson have applied AI and DL to provide technical solutions for Mobile Network Operators (MNOs) to operate and manage their networks more efficiently [1, 2].

For optimizing Capital Expenditure and Operating Expenditure with high-quality services, MNOs need a well-planned network strategy. To devise such a network strategy, MNOs should have an in-depth insight into the characteristics of their networks' behaviors. Big data generated by state-of-the-art network systems, when paired with DL techniques, can dramatically enhance network design and management [3].

In order to find insights and make important actions for mobile networks, a huge dataset must be gathered, organized, and analyzed using data analytic and mining techniques. The method of employing data analysis to predict uncertain outcomes is the subject of this work. The

objective of this research is using LSTM network to predict radio mobile networks' behaviors with real data obtained from the NetAct operational support system (OSS) system [4].

Up to now, mobile communication networks have experienced a huge change [5] which is a journey from analogue cellular systems (1G) to state-of-the-art cellular systems (5G, 6G and beyond 6G). A new generation of mobile technologies debuts roughly every ten years [6], which enhances capability, performance, effectiveness, and experience. Although each generation differentiates itself from the previous one by its standards, capacities, techniques and new features, the general network design has the common foundation of the Radio Access Network (RAN), the core network and the back-haul network [7]. These commercial mobile network technologies that are 2G, 3G, 4G and currently 5G are all connected to the MNO's OSS via signaling interfaces to operate, exploit and manage. RANs' behaviors consisting of customers' attempts, network events, major alarms, etc. are gathered to monitor, optimize and leverage network performance.

There has been extensive research on applying DL and ML to support the operation, management and exploitation of mobile networks as [8] in which a deep generative adversarial network with LSTM cells was proposed to detect anomaly KPIs in the cellular networks. In addition, it is suggested to employ a Random Forest based method to determine the primary reason for the decline in 5G radio quality [9]. Recent research has demonstrated the great potential of utilizing DL/ML to improve the quality of mobile communication networks.

Periodically, network's performance counters record and store behaviors related accessibility, retainability, mobility, etc. These counters are triggered by events generated by user equipment or network elements like base station transceivers (BTS), eNodeBs (eNBs) or serving gateway (S-GW) and so on [10]. Then, counter data which is integer values is sent to OSS's server via control planes for monitoring operation and optimization.

The 4G/LTE radio mobile network behaviors including Random Access Channel (RACH) attempts, Radio Resource Control (RRC) attempts, E-UTRAN RACH Setup attempts, etc. are logged by numerous counters. In this paper, based on the collected data, we examine the prediction of the network behaviors by using proposed LSTM network. The results has showed potential of our approach in different scenarios.

So this paper is structured as follow: In section 2, we discuss our research methodology. Section 3 presents the experiment results. Finally, section 4 concludes the paper and discusses future works.

## II. RESEARCH METHODOLOGY

### 1. The LSTM Network

The LSTM network that is an advanced form of the recurrent neural network architecture combats against the problem of vanishing gradients. The LSTM-based network is ideal for prediction of temporal sequences, and are replacing many traditional approaches to DL. In LSTM networks, each module, which is called memory cell contains three different gates. An input gate, an output gate and a forget gate denote as $i_t$, $o_t$ and $f_t$, respectively.

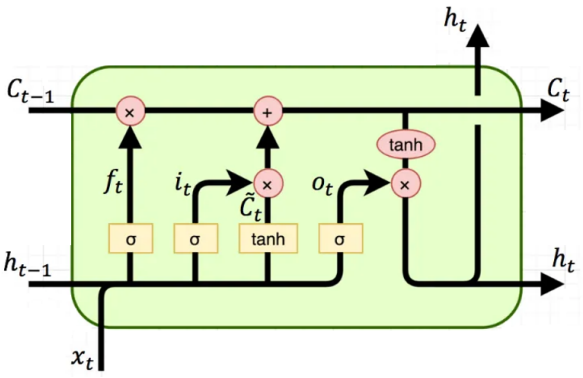



Figure 1. A basic LSTM memory cell unit [11].

The three gates in Fig. 1 are calculated as (1) to (3).

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (3)$$

Where $\sigma$ is the nonlinear activation function. The expression for an intermediate state is:

$$c_t = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (4)$$

Updates are made to the hidden state and memory cell as (5) and (6), respectively.

$$c_t = f_t \odot c_t + i_t \odot c_t \quad (5)$$

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

where $tanh$ is the nonlinear tanh activation function. The symbol $\odot$ is used to denote Hadamard product.

### 2. Implementing and Evaluating the Proposed LSTM Network

The MNO's OSS is utilized to gather the data needed to train and validate the proposed LSTM network. Since the coexisting multi-layer multi-technology mobile network has a high data volume, the amount of data that can be collected

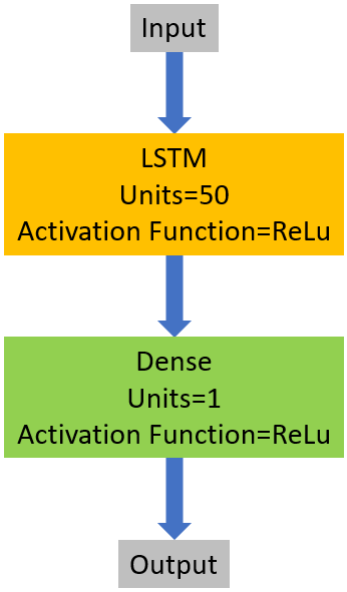



Figure 2. The proposed LSTM network's architecture.

and stored in the OSS is constrained owing to the servers' limited storage capacity. The LSTM network is trained and tested using a dataset with three different KPIs, including E-UTRAN RACH Setup Attempts, RRC Setup Attempts, and E-UTRAN Data Radio Bearer Attempts across time. It was gathered over the period of two weeks at a 15-minute granularity from one Nokia's eNodeB located in Thua Thien Hue province in Vietnam. The entire dataset which includes 1344 samples is split into two subsets with a ratio of 80% and 20% for training and testing, respectively. The network's behaviors prediction model is implemented using Python and Keras library.

A single hidden layer with 50 neurons and an output layer make up the proposed LSTM network to predict a single numerical value, as illustrated in Fig. 2. The Adaptive Moment Estimation (Adam) [12] version of stochastic gradient descent and Mean Squared Error (MSE) are used to optimize the network. Adam calculates an adaptive learning rate for each parameter and it is superior to other adaptive learning rate algorithms in terms of convergence [13].

The hidden layer and output layer are designed to utilize Rectified Linear Unit (ReLU) as the activation function in order to benefit from ReLU, which combats against the vanishing gradient problem and enhance deep neural network learning efficiency [14]. The definition of ReLU function is expressed (7):

$$f_{ReLU}(h_{i,k}) = max(0, h_{i,k}) \tag{7}$$

Other important hyper-parameters in our experiment, including Optimizer, Loss, Batch size and Epochs, are listed in Table I.

Table I
THE PROPOSED LSTM NETWORK'S MAIN HYPER-PARAMETERS.

| Hyper-parameters | Values |
|---|---|
| Optimizer | Adam |
| Loss | MAPE |
| Batch size | 50 |
| Epochs | 1000 |

Mean Average Percentage Error (**MAPE**), also known as the mean absolute percentage deviation, which is a statistical measure to define the accuracy of a learning algorithm on a specific dataset, is used as the metric to assess the prediction model's error rates across various datasets. The smaller the error is, the better the model is. If and only if the **MAPE**'s value equals zero, the model is ideal.

**MAPE** that is calculated as (8) can also be expressed in term of percentage.

$$\textbf{MAPE} = \frac{1}{p}\sum_{q=1}^{p}\left|\frac{A_q - F_q}{A_q}\right| \tag{8}$$

In (8), where $A_q$ means the actual value and $F_q$ is the predicted one. The vertical bars represent absolute values, while $p$ denotes the number of observations.

## III. EXPERIMENT RESULTS

This section provides the visualized fit-level prediction results for three 4G/LTE KPIs that describe network behaviors as from Fig. 3 to Fig. 5. As seen in these figures, our approach can predict almost all the sudden changes and overall network behavior trend which are some of the most important factors of monitoring and optimizing network operations.

As we can see, the LSTM network has a promising precision for predicting RANs' behaviors [16]. MNOs are developing and gradually transiting to a Software Defined Networking (SDN) based mobile network architecture paradigm [17]. That means they can flexibly deploy multiple mobile network technologies in the same physical elements such as radio frequency modules, antenna modules, base-band modules, etc. Therefore, thanks to forecast information related to RANs' behaviors, SDN controllers can decide to activate suitable RAN technologies to reduce cross interference and improve qualities of mobile services [18].

Moreover, investigated results of error rates of each prediction are calculated and synthesized in Table II.

Table II shows computed MAPE values of each behavior prediction:

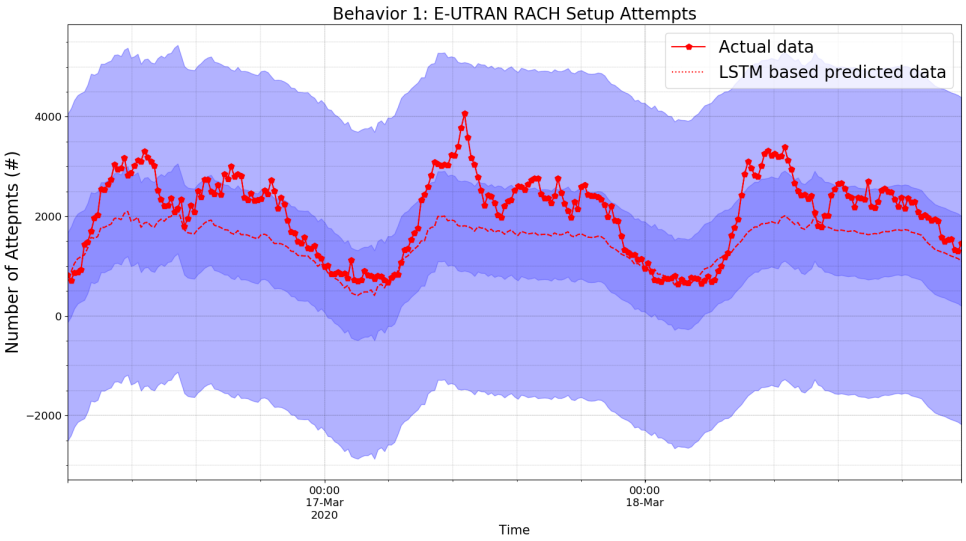



Figure 3. E-UTRAN RACH Setup attempts - predicted data versus actual data

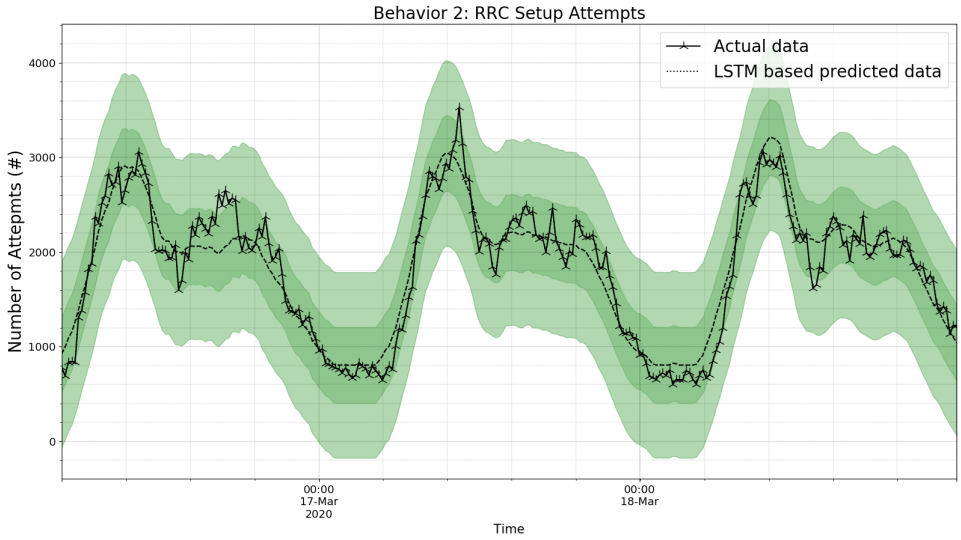



Figure 4. RRC Setup attempts - predicted data versus actual data

Table II
ERROR RATES OF EACH PREDICTION.

| Behaviors | MAPE (%) |
|---|---|
| E-UTRAN RACH Setup Attempts | 27 |
| RRC Setup Attempts | 10 |
| E-UTRAN Data Radio Bearer Setup Attempts | 12 |

According to Table II, the MAPE values varying from 10% to 27% are equivalent to the accuracy of forecasts ranging from 73% to 90%. Because of the limited training dataset retained on OSS, we can expand the dataset by utilizing data aggregation techniques to increase prediction accuracy. However, with the limited dataset, the quite high accuracy shows strong potential of our approach in predicting network behaviors.

Compared with conventional predictive models including LR and ARIMA in [18], the proposed LSTM network gives similar predictive accuracy, but the complexity of the time series dataset is more complex. The forecasting methods in [18] can only handle data with a stable and regular nature. The proposed LSTM network, on the other hand, can predict more complex data with acceptable accuracy.

Based on the proposed LSTM network, MNOs can put it in use in different prediction scenarios related to separate mobile technologies such as 3G, 4G, 5G and beyond 5G to have forecast information available for deploying appropriate technical solutions.

The operators can proactively monitor their mobile net-

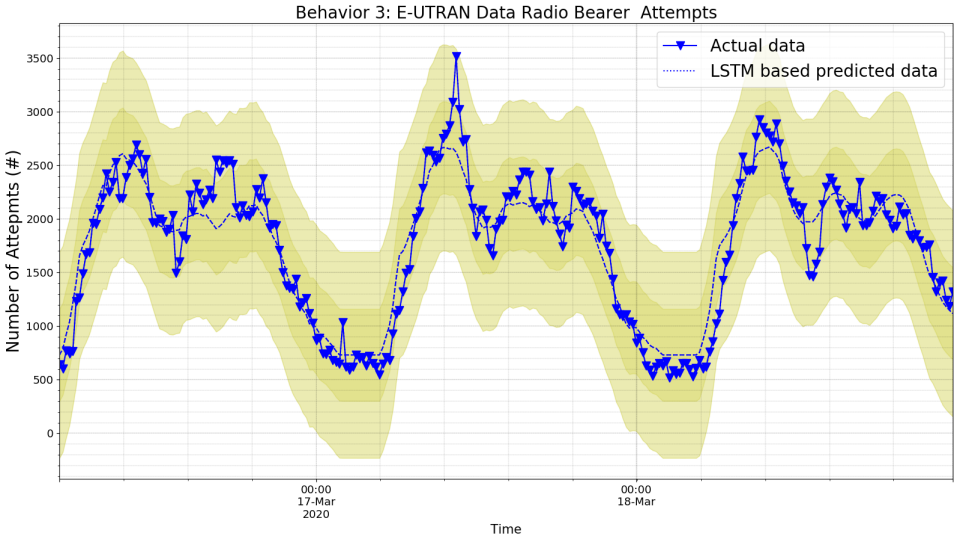



Figure 5. E-UTRAN Data Radio Bearer Setup attempts - predicted data versus actual data

works and take advantage of user-experience-driven network features by using the predicted information [2]. In addition, self-learning and adaptive networks are being researched [7] and developed to enhance qualities and capabilities of future cellular networks for satisfying subscribers' experiences.

## IV. CONCLUSION AND FUTURE WORK

Based on actual statistical data from network counters in OSS, this research proposes using the LSTM network to forecast network behaviors in radio mobile network systems. With a size-limited dataset, the model's accuracy shows its potential in forecasting a majority of counters that reflect network behaviors. With the help of network's behavior predictions, operators can have a better overview of the radio networks to allocate resources, implement technical solutions and optimize networks. In the future, we will optimize our model to improve the predicting accuracy for each network behavior and for several network behaviors together to have a better overview of the whole network.

## ACKNOWLEDGEMENT

This research was supported by Funds for Science and Technology Murata and The University of Danang - University of Science and Technology under Grant number T2021-02-07MSF.

## REFERENCES

[1] Yang Jin, Miguel Dajer, *AI-enabled Mobile Networks*, Huawei ICT, China (2019)

[2] Ericsson Page, https://www.ericsson.com/en/reports-and-papers/white-papers/artificial-intelligence-in-next-generation-connected-systems. Last accessed 4 Jan 2023

[3] I. Ahmad et al, "Machine Learning Meets Communication Networks: Current Trends and Future Challenges", *IEEE Access* ,8, 223418–223460 (2020)

[4] C. Casetti, "The 5G Ecosystem Forges Ahead as Remote Services Take Center Stage [Mobile Radio].", *IEEE Vehicular Technology Magazine*, 16(1), 7–13 (2021)

[5] A. F. M. Shahen Shah, "A Survey From 1G to 5G Including the Advent of 6G: Architectures, Multiple Access Techniques, and Emerging Technologies", *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1117–1123. IEEE, USA (2022)

[6] Qualcomm Page, https://www.qualcomm.com/media/\\documents/files/whitepaper-making-5g-nr-a-reality.pdf. Last accessed 10 Dec 2022

[7] Cuong Pham-Quoc, Tuan Tang-Anh, "Predicting mobile network's behaviors using Linear Neural Networks",*2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, pp. 75–79. IEEE, Vietnam (2021)

[8] J. Huang, E. Kurniawan and S. Sun, "Cellular KPI Anomaly Detection with GAN and Time Series Decomposition," *ICC 2022 - IEEE International Conference on Communications, Seoul, Korea*, Republic of, 2022, pp. 4074-4079, doi: 10.1109/ICC45855.2022.9838810.

[9] S. Maruyama, Y. Nishikawa, T. Onishi and E. Takahashi, "Identifying Root Cause of 5G Radio Quality Deterioration Using Machine Learning," 2022 *23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Takamatsu, Japan, 2022, pp. 1-6, doi: 10.23919/APNOMS56106.2022.9919922.

[10] Ralf Kreher, Karsten Gaenger, LTE Signaling: Troubleshooting and Performance Measurement,. 2nd ed, IEEE-Wiley Telecom, USA (2015)

[11] B. Nath Saha, A. Senapati, "Long Short Term Memory (LSTM) based Deep Learning for Sentiment Analysis of English and Spanish Data",*2020 International Conference on Computational Performance Evaluation (ComPE)*, pp.442–446. IEEE, India (2020)

[12] Diederik Kingma, P. Jimmy Lei Ba, "Adam: A Method for Stochastic Optimization", *ICLR Conference*, pp. 1–13. USA (2015)

[13] Yixiang Wang, Jiqiang Liu, Jelena Misic, Vojislav Misic, B. Shaohua Lv, Xiaolin Chang, "Accessing Optimizer Impact

on DNN Model Sensitivity to Adversarial Examples", *IEEE Access* ,7, 152766–152776 (2019)
[14] H. Ide, T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization", *In: 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2684–2691. IEEE, USA (2017)
[15] Andrew Ng, *Machine Learning Yearning: Technical Strategy for AI Engineers In the Era of Deep Learning*, Deep Learning AI, USA (2018)
[16] Amit Sharma, "Nokia Get Together NetAct ML demos", *Conference on Mobile Network Optimization*, USA (2017)
[17] Tomovic, S., Pejanovic-Djurisic, M., Radusinovic, I, "SDN Based Mobile Networks: Concepts and Benefits", *Wireless Pers Commun*, 77(4), 1629–1644 (2014)
[18] C. Pham-Quoc, H. Nguyen-Le, C. Tang-Tan, T. Tang-Anh, "Reinforcing mobile network planning with forecasted input data obtained by a machine learning approach", *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, pp. 561–565. IEEE, Vietnam (2021)

**Cuong Pham Quoc** received the B.Eng. degree in Electronic and Telecommunication Engineering from the University of Danang, University of Science and Technology, (DUT), Vietnam, in 2016. In October 2016, he joined the MobiFone Testing and Maintenance Center, MobiFone Coporation. In 2019, he received the M.Eng. degree in Electronic Engineering from the University of Danang, University of Science and Technology, (DUT), Vietnam. His research interests include mobile communications, radio network optimization, machine learning and deep learning.
Email: cuong.phamquoc@mobifone.vn

**Tuan Tang Anh** received his B.E. degree in electronic and communication from Danang University of Science and Technology, Vietnam, in 2013 and his Ph.D. degree at University of Leeds, United Kingdom, in 2019. His research interests include SDN, security, intrusion detection, machine learning and deep learning.
Email: tanganhtuan@dut.udn.vn

**Chien Tang Tan** received his Bachelor degree in Physics - Electronics from the University of Science, Ho Chi Minh City, Vietnam, in 1979. He joined the Department of Electronic and Telecommunication Engineering, the University of Danang, University of Science and Technology. In 2003, he received his Ph.D. Eng. degree in Telecommunications Engineering from Hanoi University of Technology, Vietnam. He was promoted to Associate Professor in 2010. He is currently senior lecturer at the Faculty of Computer and Electronics Engineering, Vietnam – Korea University of Information and Communication Technology, The University of Danang, Vietnam. His research interests include electromagnetic compatibility (EMC) and signal processing.
Email: ttchien@ac.udn.vn

# A Knowledge-based Framework for Developing Smart Interfaces for Smart Service Systems

Minh Truong Nguyen[1], Thang Le Dinh[2], Thi My Hang Vu[1,2]

[1] VNUHCM-University of Science, Ho Chi Minh City, Vietnam

[2] Université du Québec à Trois-Rivières, Québec, Canada

Correspondence: Thi My Hang Vu, vtmhang@fit.hcmus.edu.vn

Received: 15/03/2023, revised: 21/06/2023, accepted: 30/06/2023

Digital Object Identifier:: 10.32913/mic-ict-research-vn.2023.n1.1219

**Abstract:** Nowadays, smart service systems are value co-creating configurations of people, technologies, organisations, and information that are capable of in-dependent learning, adaptation, and decision-making. They are propelled by unprecedented advancements in connectivity, sensors, data storage, computation, and artificial intelligence. One of the key challenges faced by those systems is how to provide smart interfaces, which can assist business users with limited knowledge in business analytics in gaining business insights from business data. For this reason, this paper proposes a knowledge-based framework for developing smart interfaces for smart service systems, which will assist business users in exploring business data to gain business in-sights and subsequently make better business decisions to promote value co-creation. A prototype with simulation data has been developed and presented as a running example to illustrate how the proposed framework can be applied to create an effective smart interface for a typical smart service system: a customer intelligence system.

**Keywords:** *Smart interface design, smart service systems, customer intelligent systems, service-oriented approach.*

---

**Tiêu đề:** **Khung hỗ trợ dựa trên tri thức cho phép triển khai giao diện thông minh cho các hệ thống dịch vụ thông minh**

**Tóm tắt:** Hiện nay, các hệ thống dịch vụ thông minh nhằm tạo ra giá trị kinh tế bằng cách kết hợp con người, công nghệ, tổ chức và thông tin, có khả năng học, thích ứng và ra quyết định độc lập. Chúng được thúc đẩy bởi sự tiến bộ chưa từng có trong kết nối, cảm biến, lưu trữ dữ liệu, tính toán và trí tuệ nhân tạo. Một trong những thách thức quan trọng mà các hệ thống này đối mặt là cung cấp giao diện thông minh, giúp nhân sự trong các tổ chức doanh nghiệp có thể khai thác trí tuệ kinh doanh một cách phù hợp với vai trò và yêu cầu mà không cần hiểu biết quá sâu về kỹ thuật và các công nghệ. Vì lý do này, bài báo này đề xuất một khung hỗ trợ để phát triển giao diện người dùng thông minh cho các hệ thống dịch vụ thông minh, giúp người dùng kinh doanh khám phá dữ liệu kinh doanh để nhận thức thông tin kinh doanh và sau đó đưa ra quyết định kinh doanh tốt hơn để thúc đẩy sự tạo giá trị chung. Khung hỗ trợ sẽ cung cấp một kiến trúc tổng quát định nghĩa các thành phần thiết yếu của một hệ thống dịch vụ thông minh có hỗ trợ giao diện thông minh. Kiến trúc này bao gồm một số thành phần hệ thống đã được xây dựng sẵn và thuật toán dự đoán thành phần nào của giao diện phù hợp nhất với thông tin người dùng. Việc cung cấp các thành phần sẵn có này giúp các doanh nghiệp có thể tái sử dụng để triển khai giải pháp giao diện thông minh của chính họ với chi phí thấp nhất. Khung hỗ trợ cũng đã được tích hợp vào một hệ thống dịch vụ thông minh có sẵn nhằm minh hoạ khả năng có thể đưa vào sử dụng trong thực tế của khung hỗ trợ.

**Từ khóa:** *Thiết kế giao diện thông minh, hệ thống dịch vụ thông minh, hệ thống trí tuệ khách hàng, hướng tiếp cận hướng dịch vụ.*

---

## I. INTRODUCTION

Nowadays, smart service systems are value co-creating configurations of people, technologies, organizations, and information that are capable of independent learning, adaptation, and decision-making. They are propelled by unprecedented advancements in connectivity, sensors, data storage, computation, and artificial intelligence [1–3].

One of the key challenges faced by those systems is how to provide smart interfaces, which can assist business users with limited knowledge in business analytics in gaining business insights from business data. The term “smart interface” or “intelligent interface” refers to both the design of user interfaces for smart systems and the design of user interfaces, which utilizes intelligent approaches [4].

For this reason, this paper addressed this challenge by proposing a knowledge-based framework for developing smart interfaces for smart service systems. Smart interfaces will assist business users with a limited knowledge in business analytics in exploring business data to gain business insights and subsequently make better business decisions to promote value co-creation. Furthermore, a prototype with simulation data has been developed and presented as a running example to illustrate how the proposed framework can be applied to create an effective smart interface for a typical smart service system: a customer intelligence system.

The rest of the paper will be structured as follows. The **Theoretical Background** section introduces fundamental concepts of customer intelligent systems and smart user interfaces. The **Research Motivation** section provides an explanation for the rationale behind this paper. The **Related Work** section provides an overview of previous research conducted in the domain of smart interfaces. The **Methodology** section outlines the processes and principles employed to conduct this study. The **Framework** section describes the knowledge-based framework for the design of smart interfaces for smart service systems, including the data, information, and knowledge modules. The evaluation activities of the proposed approach is presented in the **Validation** section, which is based on a running example extracted from a specific case of an effective smart interface for a customer intelligence system, which is as a typical and popular smart service system. Finally, the **Conclusion** section will summarize the contributions of the paper and outline potential directions for future research.

## II. THEORETICAL BACKGROUND

### 1. Smart Service System

A smart service system (SSS) is a service system that can learn, adapt dynamically, and make decisions depending on information it receives, transmits, or processes to better handle a scenario in the future [1, 2]. An example of a smart service system is one that has smart devices installed to help people recognize, learn from, adapt to, monitor, and make decisions [3]. The area of smart service systems is indeed a multi-disciplinary research field that encompasses various disciplines, making it difficult to provide a comprehensive definition [1]. From the service-oriented and data-application perspective, a smart service system can be defined as a system that leverages data analytics and user interaction to assist business users in gaining valuable insights and then optimizing their service offerings [5].

In particular, customer intelligent systems (CIS) are a type of smart service systems that focus on gathering, processing, and analyzing large volumes of customer data (e.g., customer behaviors, preferences, profiles). Customers are a gold mine for valuable information and can aid in developing important business strategies [6]. By employing customer analytics, a CIS provides business users with precious insights related to their customers' needs and behaviors, that can aid them in improving customer satisfaction, increasing revenue, and achieving sustainable long-term growth.

### 2. Smart User Interface

Smart user interfaces (SUI), a sub-field of Human-Computer Interaction (HCI), focus on enhancing interaction between end-users and systems by proposing interface elements, along with relevant information to end-users intelligently [7]. Unlike traditional HCI, which focuses mainly on how to design interfaces to facilitate user actions, SUIs go further by assisting users in performing actions with incorporated knowledge.

SUIs not only enable users to perform "smart" actions but also facilitate intelligent and fashionable interaction between humans and computers by dealing with various input and output mechanisms. SUIs employ a variety of techniques and technologies, including machine learning, natural language processing, computer vision, and robotics, to create intelligent interfaces that can adapt to the user's needs and preferences. These interfaces can be found in various applications, including mobile devices, smart homes, wearable devices, and virtual and augmented reality environments.

## III. RESEARCH MOTIVATION

The paper aims at proposing a knowledge-based framework for the design and implementation of smart interfaces for SSS that will assist business users with limited knowledge in business analytics in exploring business data to gain business insights to make better business decisions.

Studies about human-centered service systems are rapidly emerging as a highly promising approach for transforming service systems into smart service systems. A smart service system can act on data analysis results to develop insights related to human behaviors to support the actions of smart service systems [1]. Smart user interfaces (SUIs) aim at facilitating intelligent interaction between humans and computers [7]. Through the use of a variety of intelligent techniques and technologies, including machine learning, natural language processing, computer vision, and robotics. Consequent, smart interfaces can be created to adapt to the user's needs and preferences that can be found in various

applications such as mobile devices [8, 9] and smart homes [10].

The topic of smart service systems and smart user interfaces has emerged in their respective domain in recent years. However, a unified approach that takes into account smart interfaces from the perspective of smart service systems is still lacking. In fact, a smart interface can assist business users in gaining business insights from business data to make better decisions and improve the service experience to promote value co-creation [1]. However, in an enterprise, business users in different roles may require varying levels of knowledge, as well as typical manners of knowledge consumption. A general interface might not be able to satisfy all the needs of different user roles. In addition, the emergence of big data has created both opportunities and challenges for exploring business data from large data sources [11, 12]. For instance, big data is a great source for promoting different types of customer intelligence, such as customer segmentation based on their characteristics, customer journey discovery, product/service recommendation, or customer value calculation [13, 14]. On the other hand, information overload may make it difficult for business users to benefit from smart service systems.

To respond to this issue, smart interfaces should be designed in such a way that they can learn about the context of a business user (e.g., needs, profiles, behaviors), and then present the user with appropriate interface elements and pertinent information to facilitate their decision-making process.

Consequently, this paper focuses on analyzing and then proposing a knowledge-based framework for answering the following research question:

*"How to design and implement a smart user interface for a smart service system?"*.

## IV. RELATED WORK

Various studies have explored the design and the implementation of intelligent user interfaces, while others emphasize the critical role that smart user interfaces plays in the success of business intelligence and recommendation systems [15]. The next paragraphs provide an overview of earlier studies done in the field of smart interfaces.

For example, there is a study that aimed to improve customer satisfaction through optimizing the user interface [16]. The study has developed a conceptual model that enables the classification of different levels of the user interface design in the field of business analytics (e.g., Visual Friendly UI for basic users, Advanced UI for experienced users, and Professional UI for pro-users).

Another study addressed the challenges of smart user interface design based on an analysis of the characteristics of users and their emotional state [17]. The system will create user groups with appropriate interface design based on this analysis.

Thus, a study based on a rule-based adaptation mechanism is proposed for the educational sector [18]. By using rules, it is possible to identify user interface components and educational content that are tailored to the individual needs of learners.

Finally, the SitAdapt system was presented that enables context-aware adaptations of the user interface during design and runtime [19]. The system is currently operational in the domains of e-business and travel and incorporates a variety of situation patterns to facilitate the adaptation process.

Although numerous studies have acknowledged the importance of smart user interfaces and focused on the smart interface design and implementation, none has yet proposed a comprehensive framework for developing such interfaces for a smart service system in the context of business analytics. Based on the knowledge-based perspective [2, 20], the proposed framework is one of the first that aims at assisting business users with limited knowledge in business analytics to transform business data into business information and explore that information to gain business insights.

## V. METHODOLOGY

The exploratory research methodology is used to conduct the study (see Figure 1).

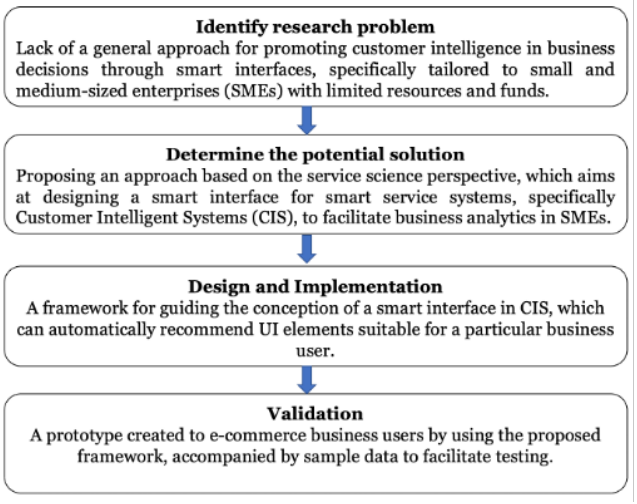



Figure 1. Adopted Methodology

The methodology includes a literature review to identify the research problem and determine the potential solution, experimentation to design and implement the proposed smart interface, and a validation to evaluate the proposed

approach [21]. To demonstrate the design and implementation of the proposed framework, a running example extracted from a specific case of an effective smart interface for a typical smart service system: a customer intelligence system will be presented accordingly in the rest of the paper. Indeed, customer intelligence systems have been selected because those systems can help business users, especially marketers, to acquire knowledge and skills from big data through customer analytics and then apply them to the process of creating, communicating, delivering, and co-creating to promote customer intelligence [14].

## VI. A KNOWLEDGE-BASED FRAMEWORK FOR SMART INTERFACE

This section presents a comprehensive knowledge-based framework for developing smart interfaces for smart service systems that considers the specific needs and preferences of business users, including essential elements of the framework. It can serve as a guide for the conception and implementation of future SSS projects that promote smart user interfaces. The overall architecture of the framework comprises a front-end layer, a backhand layer, and data sources (see Figure 2).

The **front end layer** serves as the interface between users and the SSS and is responsible for visualizing UI elements. In the fronted layer, the elements such as dash-boards, reports, or charts are designed and presented interactively to provide an intuitive user experience in order to be adaptable to a particular business user profile.

The **back end layer** handles data management, analytics, as well as UI recommendations. To structure the elements of the back end layer, the study has drawn inspiration from the DIKW model [20] and organized these elements hierarchically into three levels: Data management, Information management, and Knowledge management. The back end layer is the core of the framework and will be presented in more detail in the next paragraphs.

Concerning the **data sources**, the back end layer gathers, organizes, and conducts analytics on business data from a variety of data sources, including websites, enter-prise databases, and even the internet. For instance, related to customer intelligence systems, the collected data can be categorized into two types: customer data and business user data. The former contains information about customer's demographic, transactions, or customer interactions, which is examined to produce business insights. The latter, on the other hand, comprises business user profiles and business user history data, which is utilized to recommend UI elements.

In summary, the back end layer is designed in a manner that enables the smooth transition of data, information, and knowledge between different modules of the system. The primary focus of this paper is to utilize (business) user data to create intelligent user interfaces. Analyzing business data is beyond the scope of this paper and will not be explored. More information about the suggested architecture is shown in Figure 2, whereas the next subsections present the architecture in more detail.

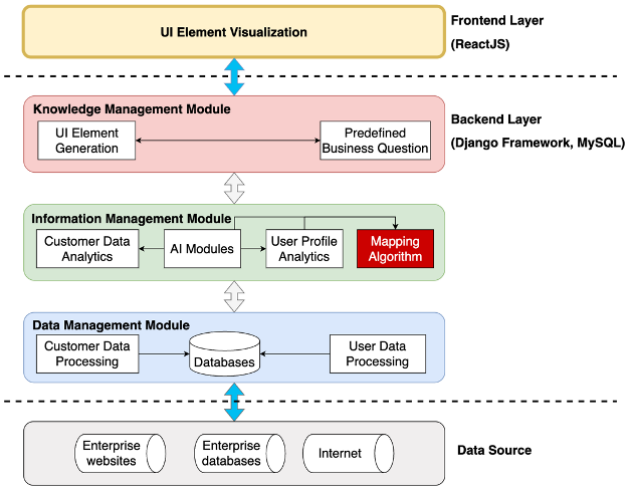



Figure 2. General architecture of the proposed framework

### 1. Data Management Module

The Data management module is composed of two key elements for the collection, processing, and transfer of customer data (**Customer data processing**) and business user data (**User data processing**) into the systems' databases. To structure the data, a tabular-based organization is used, which is suitable for the mapping algorithm in the superior layer. For UI element generation, (business) user profile and history are considered in this project. On one hand, user profiles describe specific characteristics of users, including their gender, location, level of management, industry, and any pertinent business questions or reports. These profiles are utilized to customize user interfaces based on characteristics. On the other hand, the user history comprises details about the user's previous interactions, preferences, and behavior over time. By tracking user history and analyzing previous behavior data, it is possible to enhance recommendations for UI elements, improving the user's experience.

### 2. Information Management Module

The Information management module consists of four elements, including artificial intelligence (AI) modules, Business data analytics, User profile analytics, and Mapping algorithm.

Firstly, the **AI modules** are pre-programmed with data mining and machine learning techniques such as linear regression, K-nearest neighbors (KNN), random forest, decision tree, and LightGBM, which are employed to discover relationships among data. Secondly, in the **customer data analytics**, AI modules are used for grouping similar customers, recommending products, and predicting profitable products/customers (a summary of those use cases can be found in [14]). Thirdly, in the user profile analytics, recommendation techniques, such as collaborative filtering, are applied to analyze the profiles and historical data of business users. These techniques are used to predict the most suitable user interface (UI) elements for a particular user profile.

Finally, the **mapping algorithm** is the key element for smart user interface recommendations. The algorithm receives a business user's profile specifying role (e.g., managers, staffs), domain (e.g., IT, marketing), and fields of interests (e.g., customer engagement and social media) as input data (see Figure 4)). Following that, the algorithm is used to calculate scores for UI elements based on each user's profile. These scores are used to identify the most suitable UI elements for a specific business user. The interaction between business users and UI elements is also collected each time they perform an action on a UI element to improve the scores. The output data of the algorithms consists of a collection of the most suitable UI elements, along with relevant data.

The mapping algorithm consists of the following main steps (see Figure 3):

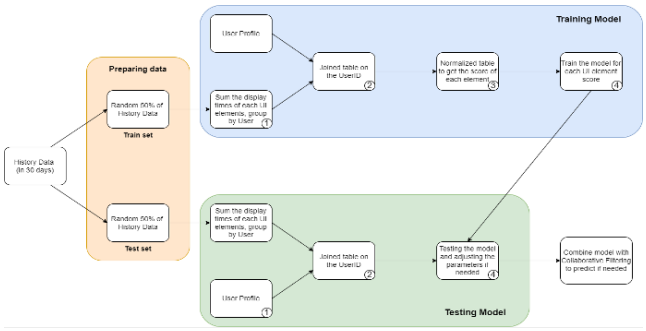



Figure 3. Mapping algorithm

- Firstly, the most recent data from the databases is extracted. In this study, data will be collected from the previous 30 days. This data was then divided into training (50%) and test datasets (50%) randomly to train and evaluate our model.
- In the next step, the usage frequency of each UI element for every user is summarized. This data will be combined with the user profiles through a jointure process, resulting in a complete dataset for analysis.
- The number of occurrences of each UI element for a specific user will be normalized before being sent to the training model.
- Several training models based on machine learning techniques, such as KNN, Random Forest, can be used to predict the score of each UI element for a new user profile.
- The last step (optional) based using the collaborative filtering algorithm (i.e., a very popular technique used to recommend similar products according to behaviors of similar users) can be performed to improve the accuracy of predicted scores.

## 3. Knowledge Management Module

The Knowledge Management module comprises two key components: UI element generation and a **predefined set of business questions**, which is a collaborative result of Dimalab, Canada, as proposed in 2022 (see Table I).

Table I
LINKING OF BUSINESS QUESTIONS, UI ELEMENTS, AND PERTINENT DATA

| Business questions | UI Elements | Pertinent Data |
|---|---|---|
| Which products can be sold together? | Top Products, Highlighted Products, Best Sales | productName, productDescription, productCategory |
| Which customers have relations to others? | Referred Customers, Best Referrals | visitNumber, sessionId, visitorId |
| Who are the most profitable customers? | Revenue from Top Customers | transactionRevenue, productId, customerId |
| How the website traffics? | Line chart of website traffics | visit, timeOnSite, hits, screenView, uniqueScreenViews |
| How attraction of website to new customers? What is the turn-back rate? | Bar chart of new user creation and visit back | lastVisitDate, createdAt |
| Where do visitors come from? | Pie chart of sources of the visitors | trafficSources |

**UI element generation** utilizes the results of the mapping algorithm (scores as-signed to UI elements for each user) in the Information management module to generate a smart user interface consisting of the recommended UI elements. The generated interface includes relevant data, which is sent to the front end layer for visualization when the user logs into the system. The recommended UI elements will be arranged according to the scores assigned

to each element. For example, the highest-scoring UI element will be displayed at the top of the page, indicating its importance and relevance. Returning to the running example about customer intelligence systems, *"What is the most profitable customers"* could be addressed through a bar chart associated with revenue from customers. Table 1 presents the relationships between predefined business queries, user interface (UI) components, and pertinent data.

## VII. VALIDATION

The proposed approach was implemented and validated by developing the smart interface prototype for an open-source customer intelligence system for small and medium enterprises (SMEs), along with some sample datasets for testing the prototype. In-deed, business users in SMEs have a limited knowledge and expertise in information technology in general and business analytics, in particular [14] that are a real-world situation for validation of the framework.

In terms of tools and technologies for developing this test prototype, Django framework (www.djangoproject.com), which is a very well-known and powerful plat-form for data analytics and machine learning algorithms is selected for the back end. Furthermore, MySQL (www.mysql.com), which is a highly suitable database for storing a moderate amount of data for this test prototype is also selected for the back end layer implementation. Concerning the front end layer, ReactJS (reactjs.org) was selected because of its robust capabilities in creating dynamic and engaging interfaces.

A possible testing scenario could be outlined as follows:

1) A business user A creates a new account and provides their profile information (see Figure 4).
2) The mapping algorithm allows assigning a score to each UI element, such as the top product bar chart or website traffic line chart, based on the similarity of user A's profile data to existing data associated with those elements.
3) A predefined threshold value is used to determine which UI elements are suitable for the user A based on the UI element's score.
4) If a UI element has a score greater than the threshold value, it is considered suitable and will be automatically recommended to the user whenever they log in to the system (see Figure 5).
5) User A's interaction with UI elements is also used as feedback, which is stored in the history data to recalculate UI element scores. This will help improve the accuracy of the mapping algorithm over time.

Figure 4 depicts a piece of user profile data, including business roles (e.g., director, manager, staff), demographics (e.g., gender, location), and other relevant information, while Figure 5 expresses the UI elements that are dynamically generated based on the profile of User A.

| user_id | gender | location | domain | seniority | interested_field_market_and_product | interested_field_business_and_revenue | interested_field_target_marketing | interested_field_positioning_and_differentation | interested_field_customer_engage_and_social_media |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Male | RE | Industrial Ma | staff | 0 | 0 | 1 | 1 | 0 |
| 2 | Female | TH | n/a | manager | 0 | 1 | 0 | 0 | 1 |
| 3 | Male | CN | Auto Parts:O | staff | 0 | 1 | 1 | 1 | 0 |
| 4 | Male | JP | Auto Parts:O | manager | 1 | 0 | 1 | 0 | 0 |
| 5 | Female | CN | Semiconduct | director | 0 | 1 | 1 | 1 | 1 |
| 6 | Male | CN | Major Pharm | leader | 0 | 0 | 0 | 1 | 0 |

Figure 4. User profile data sample

For example, tabular representations can list top products, while charts (bar charts, pie charts, and line charts) can illustrate website traffic (see Figure 5). The generated UI elements, reconfigured with relevant data, assist business users in obtaining valuable insights. This, in turn, can help them answer critical business questions, such as *"What are the most profitable products? How does the website traffic? Or where are visitors coming from? etc."*.

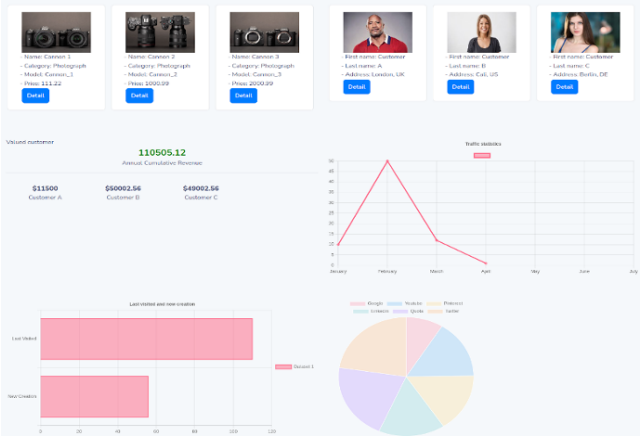



Figure 5. Example of UI elements generated for a manager

Table 2 provides a brief comparison of our framework to other relevant studies in the literature. Each paper will be evaluated based on 5 units of analysis (UA), or aspects to be investigated: *Data Management (UA1), Information Management (UA2), Knowledge Management (UA3), Transition between Data, Information, & Knowledge (UA4), and Integration into CIS (UA5)*. As shown in the table, our proposed framework covers all critical phases of data, information, and knowledge management, from initial handling to final integration into a CIS system. Furthermore, our framework places a strong emphasis on facilitating the smooth transition of data, information, and knowledge. This integration improves the framework's ability to function effectively in a variety of technological environments. SMEs can optimize and reuse the framework to create or integrate a new one into their existing system at a low cost.

Table II
COMPARISON TABLE

| Paper | UA1 | UA2 | UA3 | UA4 | UA5 |
|---|---|---|---|---|---|
| [16] | x | x | | | |
| [17] | x | x | | | |
| [18] | x | x | | | |
| [22] | x | x | | x | |
| [23] | x | x | | | |
| This study | x | x | x | x | |

## VIII. CONCLUSION

The topics of smart user interfaces have been investigated in the field of human-computer interface (HCI) for many years. However, there has been limited research exploring smart interfaces from the perspective of smart service systems. This study is among the first to combine the principles of smart service systems and user inter-face design to create an interface that is user-friendly, intuitive, and customized to the specific needs of business users.

In terms of implication, by integrating these domains, the study aims at developing an interface that can effectively analyze business data, present it in a meaningful way, and provide insights to enhance the productivity and performance of business users, leading to improved business outcomes.

Since business users of those systems usually have limited capacity to use analytics tools, a smart interface is critical to provide an opportunity for business users with limited knowledge and experience to adopt analytics tools, which may help them to maintain the company's competitive position and generate positive results with minimal costs and efforts. In addition, the prototype implementation provided a practical example of how the suggested approach can be utilized to create an effective smart user interface for customer intelligence systems, which is a typical and popular smart service system.

In terms of future work, there are various potential research areas that can be explored to continue the study. Firstly, the algorithm used in the system is only suitable for a small amount of data, and it may require significant effort to process data when user data grows fast. Secondly, the current prototype is primarily concerned with recommending UI elements, but interesting future research could explore recommending data elements that align with the needs of specific business users. Finally, exploring the business aspects of smart interfaces to demonstrate how smart inter-faces can address specific business questions should be also a priority for future work.

## REFERENCES


[1] C. Lim and P. P. Maglio, "Clarifying the concept of smart service system," *Handbook of Service Science, Volume II*, pp. 349–376, 2019.

[2] T. Le Dinh, T. T. Pham Thi, C. Pham-Nguyen, and L. N. H. Nam, "A knowledge-based model for context-aware smart service systems," *Journal of Information and Telecommunication*, vol. 6, no. 2, pp. 141–162, 2022.

[3] D. Beverungen, C. F. Breidbach, J. Poeppelbuss, and V. K. Tuunainen, "Smart service systems: An interdisciplinary perspective." *Inf. Syst. J.*, vol. 29, no. 6, pp. 1201–1206, 2019.

[4] E. M. Roth, J. T. Malin, and D. L. Schreckenghost, "Paradigms for intelligent interface design," in *Handbook of human-computer interaction*. Elsevier, 1997, pp. 1177–1201.

[5] C.-H. Lim, P. P. Maglio, K.-J. Kim, M.-J. Kim, and K.-H. Kim, "Toward smarter service systems through service-oriented data analytics," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE, 2016, pp. 936–941.

[6] J. Hagel III and J. F. Rayport, "The coming battle for customer information," *The McKinsey Quarterly*, no. 3, p. 64, 1997.

[7] J. Yven and H. Wechsler, "Smart interfaces for human-computer intelligent interaction," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, vol. 2. IEEE, 2003, pp. 1192–1197.

[8] A. Braham, F. Buendía, M. Khemaja, and F. Gargouri, "User interface design patterns and ontology models for adaptive mobile applications," *Personal and Ubiquitous Computing*, pp. 1–17, 2021.

[9] M. W. Iqbal, N. A. Ch, S. K. Shahzad, M. R. Naqvi, B. A. Khan, and Z. Ali, "User context ontology for adaptive mobile-phone interfaces," *IEEE Access*, vol. 9, pp. 96 751–96 762, 2021.

[10] G. Santos, T. Pinto, Z. Vale, and J. M. Corchado, "Multi-agent semantic interoperability in complex energy systems simulation and decision support," in *2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)*, 2019, pp. 1–6.

[11] N. A. K. Dam, T. Le Dinh, and W. Menvielle, "Towards a conceptual framework for customer intelligence in the era of big data," *International Journal of Intelligent Information Technologies (IJIIT)*, vol. 17, no. 4, pp. 64–80, 2021.

[12] B. Kitchens, D. Dobolyi, J. Li, and A. Abbasi, "Advanced customer analytics: Strategic value through integration of relationship-oriented big data," *Journal of Management Information Systems*, vol. 35, no. 2, pp. 540–574, 2018.

[13] N. A. K. Dam, T. L. Dinh, and W. Menvielle, "The quest for customer intelligence to support marketing decisions: A knowledge-based framework," *Vietnam Journal of Computer Science*, vol. 9, no. 03, pp. 349–368, 2022.

[14] T. Le Dinh, T. M. H. Vu, N. A. K. Dam, and C. N. Nguyen, "Trivi: A conceptual framework for customer intelligence systems for small and medium-sized enterprises," 2022.

[15] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda," in *Proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–18.

[16] S. A. Pednekar and S. Chandna, "Optimizing business sales and improving user experience by using intelligent user interface."

[17] T. Zubkova and L. Tagirova, "Intelligent user interface design of application programs," in *Journal of Physics: Conference Series*, vol. 1278, no. 1. IOP Publishing, 2019, p. 012026.

[18] S. V. Kolekar, R. M. Pai, and M. P. MM, "Rule based adaptive user interface for adaptive e-learning system," *Education and Information Technologies*, vol. 24, pp. 613–641, 2019.
[19] C. Märtin, F. Kampfer, C. Herdin, and L. B. Yameni, "Situation analytics and model-based user interface development: a synergetic approach for building runtime-adaptive business applications," *Complex Systems Informatics and Modeling Quarterly*, no. 20, pp. 1–19, 2019.
[20] J. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, vol. 33, no. 2, pp. 163–180, 2007.
[21] A. Dresch, D. P. Lacerda, J. A. V. Antunes Jr, A. Dresch, D. P. Lacerda, and J. A. V. Antunes, *Design science research*. Springer, 2015.
[22] Y. Dankov, "Conceptual model of user interface design for general architectural framework for business visual analytics," in *Proceedings of the 20th International Conference on Computer Systems and Technologies*, 2019, pp. 251–254.
[23] V. Johnston, M. Black, J. Wallace, M. Mulvenna, and R. Bond, "A framework for the development of a dynamic adaptive intelligent user interface to enhance the user experience," in *Proceedings of the 31st European Conference on Cognitive Ergonomics*, 2019, pp. 32–35.

**Minh Truong Nguyen** received his Bachelor's and M.S. degrees at the University of Science - Vietnam National University Ho Chi Minh City, Vietnam (VNUHCM-US), in 2017 and 2023, respectively. He has been working as Database and Data Engineer since 2018. His research interests include Data Engineering, Database Optimization and Smart Services.
E-mail: 20C14009@student.hcmus.edu.vn

**Thang Le Dinh** received his Bachelor's degree at the University of Science - Vietnam National University Ho Chi Minh City, Vietnam (VNUHCM-US) in 1990. Subsequently, he pursued his M.S. degree at the University of Geneva, Switzerland, which he completed in 1995. In 2004, he obtained his Ph.D. degree in Information Systems from the University of Geneva. Currently, he holds the position of full professor in Management Information Systems at Université du Québec à Trois-Rivières, Canada. His primary research interests revolve around various areas including service modeling, knowledge management, enterprise systems, e-business, e-collaboration, e-learning, and project management. In particular, his recent research has been focused on the global management, science, and engineering of services and information systems.
E-mail: thang.ledinh@uqtr.ca

**Thi My Hang Vu** received her Bachelor's degree at the University of Science - Vietnam National University Ho Chi Minh City, Vietnam (VNUHCM-US), and her M.S. degree at the University of Toulouse II, France, in 2008 and 2011, respectively. She further pursued her Ph.D. degree in Information Technology at the University of Grenoble Alpes, France, completing it in 2020. Currently, she works as a lecturer at the Faculty of Information Technology, VNUHCM-US. Her research interests primarily focus on knowledge science, specifically applying knowledge extraction to information systems, with a particular focus on intelligent systems and services for educational or economic support.
E-mail: vtmhang@fit.hcmus.edu.vn

# IoT Botnet Detection and Classification using Machine Learning Algorithms

Pham Van Quan[1], Ngo Van Uc[1], Do Phuc Hao[2,3], Nguyen Nang Hung Van[4]

[1] Dong A University, Da Nang, Vietnam

[2] The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Saint-Petersburg, Russian Federation

[3] Da Nang University of Architecture, Da Nang, Viet Nam

[4] The University of Da Nang - University of Science and Technology, Da Nang, Vietnam

Correspondence: Nguyen Nang Hung Van, Email: nguyenvan@dut.udn.vn

Received: 15/3/2023, revised: 23/6/2023, accepted: 30/6/2023

Digital Object Identifier:: 10.32913/mic-ict-research-vn.v2023.n1.1221

**Abstract:** This scholarly research paper addresses the crucial and complex challenge of detecting and categorizing Internet of Things (IoT) botnets through the utilization of machine learning algorithms. The study is focused on conducting meticulous analysis and manipulation of IoT botnet data, with a specific emphasis placed on the widely acknowledged IoT-23 dataset. The principal aim is to employ widely recognized and widely-used machine learning algorithms, encompassing Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), with the purpose of effectively classifying and detecting botnets within the confines of the IoT-23 dataset. By implementing these algorithms, the paper seeks to augment our understanding of their performance and efficacy within the domain of IoT botnet detection and classification. The execution of a comparative analysis, contrasting the outcomes derived from the diverse algorithms, will furnish invaluable insights into their respective merits and constraints, thereby enabling researchers and practitioners to make informed decisions concerning the most suitable algorithm for achieving successful IoT botnet detection and classification.

**Keywords:** *Internet of Things, Supervised learning, Intrucsion detection, IoT Botnet, Machine Learning.*

---

**Tiêu đề:** **Phát hiện và phân loại tấn công mạng bằng cách sử dụng các thuật toán học máy**

**Tóm tắt:** Bài báo nghiên cứu này tập trung vào thách thức quan trọng và phức tạp của việc phát hiện và phân loại các cuộc tấn công mạng (botnet-IoT) thông qua việc sử dụng các thuật toán học máy. Nghiên cứu đã tập trung vào việc phân tích tỉ mỉ và điều chỉnh dữ liệu botnet IoT, với sự tập trung đặc biệt vào tập dữ liệu IoT-23 được công nhận rộng rãi. Mục tiêu chính của bài báo là sử dụng các thuật toán học máy được công nhận và phổ biến, bao gồm các thuật toán như cây quyết định (Decision Trees), K láng giềng gần (K-Nearest Neighbors), rừng ngẫu nhiên (Random Forests) và eXtreme Gradient Boosting (XGBoost), nhằm phân loại và phát hiện tấn công một cách hiệu quả trong khuôn khổ của tập dữ liệu IoT-23. Bằng việc triển khai các thuật toán này, bài báo nhằm nâng cao nhận thức của chúng ta về hiệu suất và hiệu quả của chúng trong lĩnh vực phát hiện và phân loại botnet IoT. Thực hiện đã phân tíchvà so sánh các kết quả thu được từ các thuật toán khác nhau, sẽ mang lại những thông tin quý giá về ưu điểm và hạn chế của chúng, từ đó giúp các nhà nghiên cứu và chuyên gia có thể đưa ra quyết định về lựa chọn thuật toán phù hợp nhất để phát hiện và phân loại tấn công mạng thành công.

**Từ khóa:** *Internet của vạn vật, vọc có giám sát, vhát hiện xâm nhập, Botnet IoT, học máy.*

---

## I. INTRODUCTION

The concept of the Internet of Things (IoT) was initially proposed by K.Ashton (1999) as a solution to the increasing number of devices that require internet connectivity. The European Union Agency for Network and Information Security defines IoT as a cyberphysical ecosystem that comprises interconnected sensors and actuators that facilitate decision-making processes. Notably, IoT is a vital technology in Industry 4.0 [1]. Industrial IoT (IIoT) represents a specialized implementation of IoT that connects engines and industrial components to enhance the productivity and performance of industrial activities. IIoT achieves this goal by providing real-time monitoring, efficient management, and control of industrial processes, assets, and operational time. Moreover, IIoT aims to reduce operational costs while

improving overall efficiency [2].

However, developing IoT systems presents significant security challenges. IoT devices often have limited security and resource management. Besides IoT systems development there also should be effective security measures to protect against cyber threats and data breaches [3].

Preventing various attacks and intrusions and protecting data is very essential. Nowadays, There are a lot of Intrusion Detection Systems (IDS) have been developed to respond to this problem. IDS has a mission to detect and report intrusion systems. In addition, IDS prevents malicious at-tacks and maintains performance during at-tacks. IDS is an important component of modern security systems[4].

An IoT botnet is a network of compromised IoT devices controlled by malicious actors for malicious activities. These botnets exploit vulnerabilities in IoT devices to gain unauthorized access and create a large interconnected network of compromised devices. Key features of IoT botnets include device diversity, vulnerability exploitation, command and control infrastructure, DDoS attacks, spamming and phishing capabilities, cryptocurrency mining, and data theft and privacy breaches. To address the threats posed by IoT botnets, it is crucial to enhance IoT device security, implement strong authentication mechanisms, regularly update software, segment net-works, and educate users about security best practices.

The impetus behind crafting a scholarly paper centered on the identification and categorization of Internet of Things (IoT) botnets through machine learning techniques arises from the exigent necessity to address the escalating perilous landscape posed by these malevolent networks. The proliferation of interconnected IoT devices has undeniably amplified the potential for botnet attacks, thereby jeopardizing the well-being of both individual users and essential infrastructures. By harnessing the power of machine learning methodologies, we can augment our proficiency in discerning and classifying IoT botnets, thus facilitating the implementation of proactive measures to forestall and alleviate their deleterious consequences. Employing machine learning algorithms empowers us to meticulously scrutinize substantial volumes of network traffic data, facilitating the identification of regular patterns and anomalies, while concurrently distinguishing normal IoT device behavior from botnet operations. By immersing ourselves in this realm of inquiry, we can contribute to the formulation of robust and efficacious defensive mechanisms, thereby fortifying the security of IoT ecosystems.

The main objective of this scholarly research paper is to address the pivotal challenge of detecting and categorizing IoT botnets through the utilization of machine learning algorithms. The study places significant emphasis on the meticulous analysis and manipulation of IoT botnet data, with a specific focus on the renowned IoT-23 dataset. The primary aim is to implement widely recognized and extensively employed machine learning algorithms, namely Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), with the intention of effectively classifying and detecting botnets within the confines of the IoT-23 dataset. The application of these algorithms serves the purpose of augmenting our comprehension of their performance and efficacy when applied to the task of IoT botnet detection and classification. The execution of a comparative analysis, juxtaposing the outcomes derived from the diverse algorithms, will yield invaluable insights into their respective strengths and limitations, enabling researchers and practitioners to make well-informed decisions pertaining to the most suitable algorithm for the successful detection and classification of IoT botnets.

## II. RELATED WORKS

In recent years, research on IoT intrusion detection has been more attention. Re-search on this topic is becoming more necessary. Based on the results from previous research, there are many solutions to develop IoT intrusion detection systems. Under-standing the strengths as well as the problems of previous research helps researchers find research directions that bring great benefits for building IDS systems. Therefore, previous studies are fundamental in providing knowledge for future development.

The proliferation of IoT devices has ushered in an escalating threat posed by IoT botnets, presenting cybercriminals with new avenues for orchestrating large-scale attacks. The imperative task of detecting and categorizing these botnets assumes critical importance in curbing the burgeoning risk. Leveraging the capabilities of machine learning techniques has emerged as a potent strategy to bolster cybersecurity measures by facilitating the identification of distinctive patterns, anomalies, and malevolent behaviors ingrained within the labyrinthine network traffic of IoT botnets. This underscores the indispensability of deploying machine learning algorithms to tackle the intricate challenges inherent in IoT botnets.

J. Hajji et al. [5] conducted a study to apply unsupervised machine learning algorithms, including K-means, PCA, and Autoencoder, to detect anomalous network traffic. A. Rahim et al. [6] employed statistical analysis to determine the most significant features and then applied several machine learning algorithms, such as DT, RF, and Naive Bayes (NB), to classify normal and malicious traffic.

S. M. Z. Islam et al. [7] devised averaging and stacking models based on Support Vector Machine (SVM), RF, and

Gradient Boosting Algorithms. Y. Li et al. [8] devel-oped a bagging model from four machine learning models based on DT, KNN, Logistic Regression (LR), and RF to classify network traffic as normal or malicious.

P. H. Do [9] proposed a feature extraction method by dividing feature sets into various classes, followed by the application of machine learning algorithms to those attribute classes. These studies utilized a range of algorithms, including some combinations, to achieve high accuracy in their classification tasks.

Several researchers have explored the potential of deep learning models to detect anomalous behavior in network traffic. Alotaibi et al. [10] developed a Convolution-al Neural Network (CNN) based model to classify network traffic as normal or malicious. Li et al. [11] utilized Long Short Term Memory (LSTM) and a Dense Neural Network to classify network traffic as normal or malicious. Similarly, Abdallah et al. [12] employed a deep learning model based on CNNs and LSTM.

Kiani et al. [13] proposed a Deep Autoencoder Neural Network to learn the normal behavior of IoT network traffic and used it to detect anomalous behavior in real-time. Additionally, Rasool et al. [14] experimented with transfer learning models such as VGG16, ResNet50, and InceptionV3. These studies suggest the potential of deep learning models in detecting anomalies in network traffic.

The IoT-23 [15] dataset, released in 2020, has been a focal point for researchers in IoT security and machine learning. It has been extensively used to develop and evaluate techniques for detecting and mitigating IoT malware infections. One notable study combined machine learning algorithms with deep learning approaches to im-prove malware detection accuracy using the dataset. Another project utilized the dataset to create a tailored intrusion detection system for IoT devices, focusing on real-time detection of anomalies and malicious activities. Additionally, researchers have used the IoT-23 dataset to assess various feature extraction methods and classification algorithms for IoT malware detection. The availability of this dataset has significantly contributed to advancements in IoT security research and the development of effective algorithms and systems to protect IoT devices from malicious activities. Ongoing exploration of the IoT-23 dataset continues to enhance the security and resilience of the expanding IoT ecosystem.

Each of these researches presents different methods for detecting and classifying cyberattacks and all have shown good results. However, some researches have not mentioned the performance of the model as well as the execution time of the model, a few others only present detection or classification.

## III. METHOD

This section will commence with a presentation of the dataset that will be used in the subsequent analysis. Preprocessing techniques will then be applied to ensure the accuracy and reliability of the data. Following that, we will delve into the process of data selection, visualization, formatting, and splitting. Our analysis will culminate with an evaluation of the effectiveness of the algorithms employed, accompanied by a comparative analysis of the outcomes. The main execution flow of this study is illustrated in Figure 1.

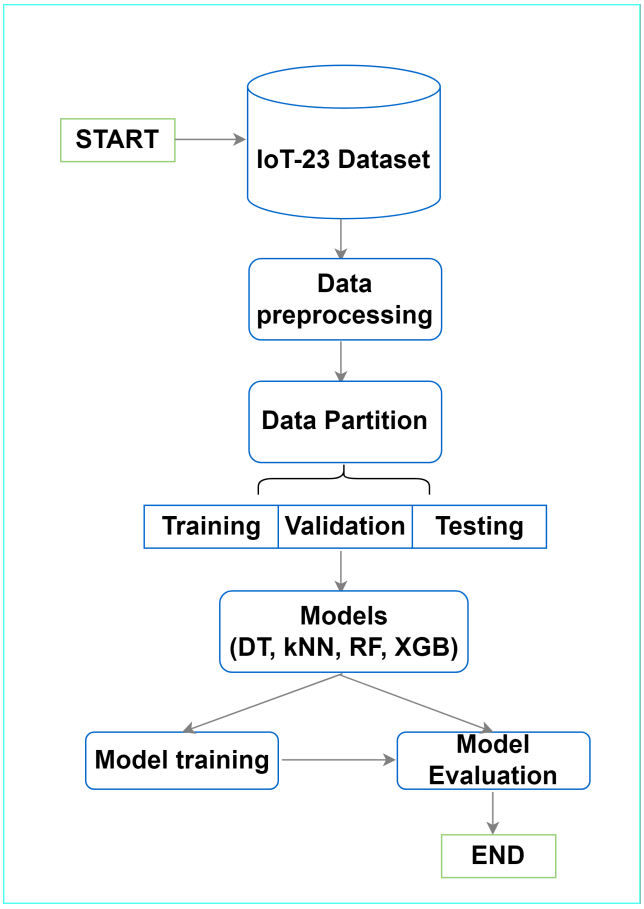



Figure 1. Proposed model

To evaluate the effectiveness of machine learning models for classifying IoT-Botnets, we first perform data preprocessing. Data preprocessing helps improve the classification process's effectiveness. Next, we divide the dataset into corresponding training, testing, and validation sets. Finally, we deploy popular machine learning models such as decision trees, k-nearest neighbors (kNN), random forests, and XGBoost (XGB) to assess the effectiveness of each model.

## 1. Dataset

The IoT-23 dataset, released in January 2020, has been a significant focus of re-search in the domains of IoT security and machine learning. Researchers have extensively utilized this dataset to develop and assess various techniques for the detection and mitigation of IoT malware infections. The dataset encompasses network traffic captures from a diverse array of 23 distinct Internet of Things (IoT) devices, spanning devices such as smart plugs, cameras, smart locks, and smart thermostats, among others. These traffic captures were meticulously obtained within a controlled environment, wherein each device was connected to a segregated Wi-Fi network and exposed to a range of attack scenarios, comprising brute-force attacks, Mirai botnet attacks, injection attacks, and more.

Table I
THE TRAFFIC CONTENT OF THE IOT-23 DATASET BY EXECUTED ATTACKS.

| Attack Name | Flows |
|---|---|
| Part-Of-A-Horizontal-PortScan | 213, 852, 924 |
| Okiru | 47, 381, 241 |
| Okiru-Attack | 13, 609, 479 |
| DDoS | 19, 538, 713 |
| C&C-Heart Beat | 33,673 |
| C&C | 21,995 |
| Attack | 9398 |
| C&C- | 888 |
| C&C-Heart Beat Attack | 883 |
| C&C-File download | 53 |
| C&C-Torii | 30 |
| File download | 18 |
| C&C-Heart Beat File Download | 11 |
| Part-Of-A-Horizontal-PortScan Attack | 5 |
| C&C-Mirai | 2 |

Detailed metadata pertaining to each capture is incorporated in the dataset, encompassing information such as device type, firmware version, and attack type. A comprehensive depiction of the traffic distribution is provided in Table I, while an elaborate listing, presenting the description of all features, is presented in Table II.

To assist in the detection of malicious traffic, the Stratosphere laboratory devel-oped labels for the different types of network flows, based on their analysis of mal-ware captures. The labels used for malicious flow detection were Attack, C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, and Torii. The "Attack" label was assigned to flows that attempted to exploit vulnerable services, such as brute-forcing a telnet login or injecting a command in the header of a GET request. The "Benign" label indicated that no malicious or suspicious activities were detected. The "C&C" label indicated that the infected device was connected to a CC server, while the "DDoS" label indicated that the infected device was participating in a Distributed Denial of Service attack. The "FileDownload" label was assigned to connections that involved the downloading of a file to the infected device.

Table II
IOT-23 DATASET FEATURES.

| Feature Name | Description |
|---|---|
| fields-ts | Start Time flow |
| uid | Unique ID |
| id.orig-h | Source IP address |
| id.orig-p | Source port |
| id.resp-h | Destination IP address |
| id.resp-p | Destination port |
| proto | Protocol |
| service | Type of Service (http, dns, etc.) |
| duration | Flow total duration |
| orig-bytes | Source-destination transaction bytes |
| resp-bytes | Destination-source transaction bytes |
| conn-state | Connection state |
| local-orig | Source local address |
| local-resp | Destination local address |
| resp-pkts | Destination packets |
| orig-ip-bytes | Flow of source bytes |
| history | History of source packets |
| missed-bytes | Missing bytes during transaction |
| orig-pkts | Source packets |
| resp-ip-bytes | Flow of destination bytes |
| label | Name of type attack |

The "Heart-Beat" label was assigned to connections that were used to track the infected host by the C&C server. The "Mirai" label was assigned to connections that exhibited characteristics of a Mirai botnet attack, while the "Okiru" label was used for connections with similar patterns to the less common Okiru botnet. The "PartOfAHorizontal-PortScan" label was assigned to connections used for horizontal port scanning to gather information for further attacks. Finally, the "Torii" label indicated connections that exhibited characteristics of a Torii botnet attack.

The availability of the IoT-23 dataset, featuring labeled malware captures and real IoT device traffic, has played a pivotal role in advancing research on IoT security. It has served as a valuable resource for the development and evaluation of machine learning algorithms, intrusion detection systems, and other security solutions aimed at safeguarding IoT devices from malicious activities. Researchers continue to explore and expand upon the insights provided by the IoT-23 dataset to further enhance the security and resilience of the rapidly expanding IoT ecosystem.

## 2. Preprocessing

The data pre-processing phase plays a crucial role in preparing the data for model training, encompassing various steps and considerations. This process, as depicted in Figure 2, is essential for ensuring data quality and addressing specific challenges that may arise during analysis.

The initial step in data preprocessing involves reading the data source, enabling access to the dataset for further examination and manipulation. Subsequently, the focus

shifts to assessing the data quality through comprehensive checks and diagnostics. This critical step aims to identify and resolve potential issues, such as missing data, outliers, or the need for numeric conversions.

To address these challenges, suitable solutions are devised for each specific issue encountered, while ensuring minimal impact on the overall dataset. This approach enables targeted interventions and safeguards the integrity of the data.

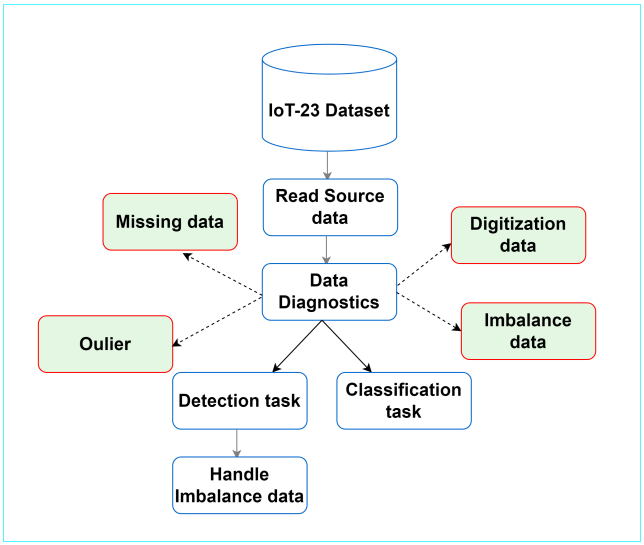



Figure 2. The steps of data preprocessing

Furthermore, it is important to acknowledge that different tasks require tailored data handling techniques. Specifically, data imbalance emerges as a prominent concern due to the substantial variations in the number of records across different classes. To mitigate this issue, distinct strategies are adopted based on the task at hand.

Upon meticulous examination and assessment of the IoT-23 dataset, it was determined that the dataset is devoid of any missing data points. However, a minor presence of outlier data points was observed. Furthermore, the analysis yielded an overview of the dataset, revealing pertinent information such as a total count of 3,000,779 records, encompassing 20 distinct features. Table III provides a comprehensive representation of the data types associated with each feature.

In order to handle features with non-numeric data types, a categorization process was conducted, resulting in the division of these features into two distinct groups. The first group comprises non-numeric data types that cannot be readily converted into a numeric format. This group includes features such as uid, id.orig_h, id.resp_h, local_orig, and local_resp. Conversely, the second group consists of non-numeric data types that can be conveniently converted into a numeric format. This group encompasses features such as proto, service, conn_state, history, duration, orig_bytes, and resp_bytes. To facilitate this transformation, each distinct value within the aforementioned features was assigned a unique integer value, starting from 0. The LabelEncoder class from the sklearn library was employed for this purpose. Subsequently, with the data type processing completed on the original dataset, experimentation was carried out on 15 select features as part of the subsequent analysis.

Table III
DATA TYPE OF IOT-23 FEATURES

| No. | Feature Name | Data Type |
|---|---|---|
| 1 | fields-ts | Float |
| 2 | uid | Object |
| 3 | id.orig-h | Object |
| 4 | id.orig-p | Float |
| 5 | id.resp-h | Object |
| 6 | id.resp-p | Float |
| 7 | proto | Object |
| 8 | service | Object |
| 9 | duration | Object |
| 10 | orig-bytes | Object |
| 11 | resp-bytes | Object |
| 12 | conn-state | Object |
| 13 | local-orig | Object |
| 14 | local-resp | Object |
| 15 | resp-pkts | Float |
| 16 | orig-ip-bytes | Float |
| 17 | history | Object |
| 18 | missed-bytes | Float |
| 19 | orig-pkts | Float |
| 20 | resp-ip-bytes | Float |
| 21 | label | Object |

In order to align the label column data type with the requirements of our research, it is imperative to undertake a bifurcation process. This division serves a crucial purpose, particularly in the context of our detection tasks. Within this framework, records bearing benign labels will be attributed a value of 0, while all other records will be assigned a value of 1. In the realm of classification tasks, classes exhibiting limited records and sharing identical attack types will be amalgamated into cohesive class entities. The intricate details of this labeling schema are meticulously outlined in Table IV, providing a comprehensive overview of the classification and detection assignments.

In the context of network environments, the occurrence of events unfolds with remarkable speed, necessitating swift, continuous, and automated data operations. Data collection is seamlessly orchestrated by IoT systems, encompassing essential parameters, and subsequently funneled into automated processing pipelines and trained models to generate real-time results. The underlying objective of these dynamic actions revolves around the prompt detection and prevention of network anomalies, ensuring timely responses and interventions.

Table IV
THE DETAILED LABEL FOR MULTI-CLASSIFICATION TASK

| Label Name | Label Number |
|---|---|
| Beingn | 0 |
| C&C-Heart Beat | 1 |
| C&C | 1 |
| C&C- | 1 |
| C&C-Heart Beat Attack | 1 |
| C&C-File download | 1 |
| C&C-Torii | 1 |
| File download | 1 |
| C&C-Heart Beat File Download | 1 |
| C&C-Mirai | 1 |
| Attack | 2 |
| DDoS | 3 |
| Okiru | 4 |
| Okiru-Attack | 4 |
| Part-Of-A-Horizontal-PortScan | 5 |

In the context of detection tasks, where distinguishing between malicious and benign classes is crucial, sampling techniques are employed to address the significant imbalance. This involves carefully adjusting the distribution of the malicious group to ensure optimal representation.

For classification tasks, where multiple classes with a limited number of records are involved, a merging approach is implemented to reduce data imbalance. This consolidation facilitates more balanced representations of the different classes, improving the classification process.

## 3. Analysis Method

*a) Decision Tree*

In the realm of IoT botnet classification, decision trees represent a foundational and effective machine learning technique. These trees offer a transparent and comprehensible framework for decision-making, leveraging the distinct attributes found within IoT network traffic data. Their extensive utilization in the field of cybersecurity demonstrates their prowess in detecting and categorizing IoT botnets.

At its core, the principle underpinning decision trees revolves around the iterative division of datasets, guided by various features, to generate a tree-like structure. Each internal node within this structure signifies a decision based on a specific feature, while each leaf node corresponds to a definitive classification outcome. Constructing a decision tree entails the meticulous selection of the most informative features and the determination of optimal splitting criteria at each node, aiming to maximize the differentiation between distinct classes.

Decision trees possess notable advantages in the context of IoT botnet classification, stemming from their ability to capture intricate relationships and feature interactions. Their versatility in handling both categorical and numerical features renders them suitable for a wide range of network traffic data. Furthermore, decision trees demonstrate robustness in the face of real-world data imperfections, accommodating missing values and outliers effectively.

The interpretability factor assumes a vital role in the realm of IoT botnet classifi-cation with decision trees. The pathways inherent to the tree structure offer invaluable insights into the characteristics and behavior of IoT botnets. Through diligent examination of the decision rules, security analysts can attain a deeper comprehension of the distinguishing features and patterns associated with different types of botnet activities.

The decision tree algorithm is a widely used and popular type of supervised learn-ing model that is effective in solving both classification and regression problems. Its structure consists of nodes that represent variables and branches that represent the relationship between these variables. At the root node, the algorithm considers the entire dataset, and through a series of binary decisions based on the input variables, it recursively partitions the data into smaller subsets. These partitions form internal nodes in the tree, and the leaves correspond to the predicted value of the target variable for each subset.

To build a decision tree, the algorithm follows a top-down approach that selects the best attribute to split the data based on a criterion such as information gain, Gini index, or entropy. Once the data is split, the algorithm recursively applies the same process to each of the resulting subsets until a stopping criterion is met, such as a predefined tree depth, minimum number of instances per leaf, or no further improvement in the predictive performance.

To update and calculate node values in a decision tree, two formulas are used: the impurity measure and the criterion for selecting the best split. The impurity measure quantifies the degree of homogeneity or heterogeneity of the target variable within a node, while the criterion for selecting the best split determines which attribute provides the highest information gain or the lowest impurity after splitting the node.

The entropy of probability distribution $\boldsymbol{p} = (p_1, p_2, p_3, \ldots, p_n)$ satisfied $\sum_{i=1}^{n} p_1 = 1$

$$H(p) = -\sum_{i=1}^{n} p_i \times \log p_i \tag{1}$$

The information gain is a node test formula that yields the amount of that node information that remains after switching to $k$ child node.

$$\text{Gain}(p) = H(p) - \left( \sum_{i=1}^{k} \frac{n_i}{n} \times H(i) \right) \tag{2}$$

*b) Random Forest (RF)*

Random Forest stands as a prominent machine learning algorithm that has exhibited remarkable effectiveness within the realm of IoT botnet classification. Its utilization as an ensemble learning technique enables the combination of multiple decision trees, culminating in predictions that are both robust and accurate. In the field of cybersecurity, Random Forest has garnered substantial adoption for its adeptness in identify-ing and classifying IoT botnets.

At the heart of the Random Forest algorithm lies the construction of an ensemble comprising decision trees. Each decision tree is trained on a random subset of the training data, incorporating a random subset of features at each node. By introducing randomness in both the data and feature selection processes, Random Forest augments the diversity within the model, effectively curbing the risks associated with overfitting. This diversification empowers the algorithm to capture varied facets and intricate patterns intrinsic to IoT network traffic data, thereby elevating the performance of classification.

Throughout the training phase, individual decision trees within the Random Forest are meticulously constructed through the recursive partitioning of data. This partitioning is conducted based on distinct feature thresholds, with the primary aim of minimizing impurity or maximizing information gain at each split. Consequently, a hierarchical arrangement of nodes and leaves materializes, serving as a collective representation of the acquired patterns and interrelationships contained within the data.

Upon reaching the prediction stage, Random Forest amalgamates the outputs generated by each individual decision tree through a voting mechanism, specifically tailored for classification. Each decision tree presents its vote for the predicted class, and the class that accumulates the majority of votes is ultimately deemed the final prediction. By leveraging this ensemble-based approach, Random Forest successfully mitigates the potential biases or errors inherent in individual decision trees, thereby fortifying the overall robustness and accuracy of classifications.

Random Forest encompasses several notable advantages in the realm of IoT botnet classification. It adeptly handles high-dimensional and intricate data, rendering it particularly suited for the analysis of the multifaceted features and patterns inherent in IoT network traffic. Additionally, Random Forest excels in estimating feature importance, furnishing invaluable insights into the pivotal factors contributing to botnet activities.

*c) K-Nearest Neighbor (KNN)*

The k-Nearest Neighbor (k-NN) algorithm stands as a significant and versatile ma-chine learning technique employed in IoT botnet classification. Esteemed for its simplicity, k-NN presents an intuitive approach to discern and classify IoT botnets based on their network traffic data.

At its core, the k-NN algorithm endeavors to assign a class to a new data point by assessing the classes of its k nearest neighbors within the feature space. This assessment relies on distance metrics like Euclidean or Manhattan distance, facilitating the identification of the k closest neighbors from the training dataset. The class label of the new instance subsequently emerges through a majority voting process conducted among these selected neighbors.

In the context of IoT botnet classification, the k-NN algorithm offers a plethora of advantages. Foremost, it exhibits the capacity to encapsulate intricate relationships and non-linear patterns prevalent in network traffic data. This adaptability empowers k-NN to accommodate diverse manifestations of IoT botnet activities and their associated features. Moreover, k-NN demonstrates proficiency in handling both categorical and numerical features, facilitating its application to the heterogeneous data en-countered in IoT environments. Lastly, the inherent simplicity of k-NN obviates the necessity for explicit training, rendering its implementation and comprehension straightforward.

However, the k-NN algorithm does harbor certain limitations. As the dataset's size expands, the computational complexity of determining the nearest neighbors escalates significantly. Consequently, the prediction times lengthen, particularly in high-dimensional spaces. Furthermore, the performance of k-NN hinges on the judicious selection of the parameter k, as an inappropriate choice can result in underfitting or overfitting.

*d) Extreme Gradient Boosting (XGBoost)*

Extreme Gradient Boosting (XGBoost) stands as a formidable machine learning algorithm that has garnered notable acclaim within the realm of IoT botnet classification. As an ensemble learning method, XGBoost seamlessly amalgamates multiple weak prediction models, typically decision trees, to construct a robust and precise classifier. Its exceptional performance and capacity to tackle intricate IoT botnet classification tasks have positioned XGBoost as a favored choice in the realm of cybersecurity.

The fundamental functioning of the XGBoost algorithm lies in its iterative training process, wherein weak prediction models are sequentially incorporated into the en-semble while concurrently minimizing the overall prediction error. Each subsequent model concentrates on capturing the residual errors of its predecessors, progressively refining the predictive capabilities of XGBoost. By skillfully combining gradient boosting and regularization techniques, XGBoost adeptly balances model complexity and general-

ization prowess, culminating in superior performance.

Within the context of IoT botnet classification, XGBoost affords several notable advantages. It effortlessly navigates the challenges posed by high-dimensional and heterogeneous data frequently encountered in IoT environments, thereby facilitating the incorporation of diverse features and patterns intrinsic to botnet activities. Additionally, XGBoost excels at capturing intricate relationships and interactions among these features, heightening its ability to accurately classify IoT botnets. Moreover, XGBoost seamlessly integrates regularization techniques that effectively mitigate the perils of overfitting, enhancing its robustness and generalization capabilities.

Nevertheless, XGBoost encounters challenges in terms of interpretability. As an en-semble of decision trees, elucidating the individual contributions of each tree within the XGBoost model can prove arduous. Nonetheless, techniques centered around feature importance analysis can be employed to gain insights into the relative significance of features in the classification process.

Extreme Gradient Boosting (XGBoost) emerges as a potent and commendable algorithm within the domain of IoT botnet classification. Through its ensemble learning approach, amalgamated with gradient boosting and regularization techniques, XGBoost adeptly captures intricate relationships and deftly handles high-dimensional data. Notwithstanding potential challenges pertaining to interpretability, XGBoost's resolute performance and accurate classification of IoT botnets underscore its indispensable role in the cybersecurity domain.

## 4. Performance Evaluation

In assessing the algorithmic models described earlier, diverse techniques were em-ployed to gauge the precision of the outcomes and derive comprehensive findings for each model. This study involved several fundamental concepts, such as TP, TN, FP, and FN. TP signifies the count of true positives that have been accurately determined, while TN is the number of true negatives that have been correctly identified. FP represents the actual number of positive cases that have been erroneously classified as negative, whereas FN denotes the count of negative cases that have been mistakenly classified as positive.

### *a) Precision (PRE)*

Precision is a performance metric utilized to assess a model's effectiveness by determining the proportion of accurately identified positive instances. This measure can be mathematically calculated through the use of a formula, which takes into account the number of true positives and false positives. Specifically, precision is computed by dividing the number of true positives by the sum of true positives and false positives.

$$PRE = TP/(TP + FP) \tag{3}$$

### *b) Accuracy (ACC)*

Accuracy is a performance metric utilized to gauge a model's efficacy by determining the proportion of correct predictions out of the total number of predictions. This measure can be mathematically calculated through the use of a formula, which considers the number of true positives and true negatives. Specifically, accuracy is computed by dividing the sum of true positives and true negatives by the total number of predictions.

$$ACC = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{4}$$

### *c) Recall Score (RE)*

The recall score is a performance metric utilized to evaluate a model's efficacy in correctly identifying actual positive instances. This measure can be mathematically calculated through the use of a formula, which incorporates the number of true positives and false negatives. Specifically, the recall score is computed by dividing the number of true positives by the sum of true positives and false negatives.

$$RE = TP/(TP + FN) \tag{5}$$

### *d) F1 Score for Binary class*

The F1 score is a performance metric that is obtained by averaging both the accuracy and recall scores. This measure is widely used in assessing the overall effectiveness of a model since it provides a comprehensive view of false positives and false negatives. Specifically, the F1 score is calculated as the harmonic mean of precision and recall. The formula for computing the F1 score takes into account both the number of true positives, false positives, and false negatives, thereby providing a more balanced evaluation of a model's performance.

$$F1 - \ \text{score} \ = 2 \times \frac{PRE \times RE}{PRE + RE} \tag{6}$$

### *e) F1 Score for Multi-class*

The F1 score [16] is a widely recognized and commonly employed metric for evaluating the performance of multi-class classification algorithms. It serves as a comprehensive measure that takes into account both precision and recall, thereby enabling a balanced assessment of the classifier's overall efficacy.

Within the context of multi-class classification in the IoT-botnet problem, the F1 score offers valuable insights into

the algorithms' capability to accurately classify instances across diverse classes. By considering the occurrence of false positives (misclassified instances) and false negatives (missed instances) for each class, the F1 score provides a robust evaluation of the algorithm's performance.

The calculation of the F1 score involves determining the harmonic mean of precision and recall for each class. This approach effectively evaluates the equilibrium between correctly identifying positive instances (precision) and capturing all positive instances (recall). A higher F1 score signifies a superior balance between precision and recall, thereby demonstrating the classifier's competence in accurately identifying instances across all classes.

To summarize, the F1 score assumes a pivotal role as a critical metric in assessing the performance of multi-class classification algorithms in the IoT-botnet problem. It facilitates a comprehensive evaluation by considering precision and recall for each class, thereby enabling a well-rounded assessment of the classifier's overall effectiveness in accurately classifying instances across multiple classes.

## IV. RESULTS AND DISCUSSION

### 1. Binary Classification

Table V presents a comprehensive analysis of the evaluation results obtained from binary classification in the IoT-botnet problem. Four distinct machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGB), were employed to discern their performance using various metrics.

Table V
SOME METRICS OF BINARY CLASSIFICATION

| Evaluation | DT | RF | KNN | XGB |
|---|---|---|---|---|
| ACC | 99.9% | 89.5% | 99.6% | 99.8% |
| PRE | 100.0% | 88.0% | 100.0% | 100% |
| RE | 100.0% | 86.0% | 99.0% | 100% |
| F1 | 100.0% | 87.0% | 100.0% | 100% |
| Time (s) | 5.9 | 40.2 | 58.2 | 99.4 |

In terms of accuracy (ACC), the Decision Tree algorithm exhibited unparalleled precision, achieving an exceptional accuracy rate of 99.9%. Close on its heels, Extreme Gradient Boosting demonstrated an impressive accuracy of 99.8%. k-Nearest Neighbor showcased commendable accuracy at 99.6%, while Random Forest achieved a respectable accuracy of 89.5%.

Precision (PRE), which measures the ability to accurately classify positive instances among all instances predicted as positive, yielded noteworthy outcomes. Both Decision Tree and Extreme Gradient Boosting achieved flawless precision scores of 100.0%. k-Nearest Neighbor also showcased excellent precision, matching the perfect score of 100.0%. Although slightly lower, Random Forest performed commendably with a precision score of 88.0%.

The evaluation of Recall (RE), which quantifies the proportion of correctly classified positive instances among all actual positive instances, revealed exemplary results. Both Decision Tree and Extreme Gradient Boosting excelled with perfect recall scores of 100.0%. k-Nearest Neighbor demonstrated a commendable recall rate of 99.0%, while Random Forest achieved a respectable recall of 86.0%.

The F1 score, representing the harmonic mean of precision and recall, provided an overall assessment of the classifiers' performance. Both Decision Tree and Extreme Gradient Boosting achieved perfect F1 scores of 100.0%. k-Nearest Neighbor showcased excellent performance with an F1 score of 100.0%, while Random Forest achieved a respectable score of 87.0%.

In terms of computational time, the Decision Tree algorithm emerged as the most efficient, requiring a mere 5.9 seconds for processing. Random Forest followed with a processing time of 40.2 seconds, while k-Nearest Neighbor demanded 58.2 seconds. Extreme Gradient Boosting exhibited the longest processing time, clocking in at 99.4 seconds.

### 2. Multi classification

Table VI illustrates the comprehensive evaluation and analysis results pertaining to multi-classification in the context of the IoT-botnet problem. The study involved the assessment of four distinct machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGB), utilizing a range of performance metrics.

The primary metric of accuracy (ACC) reveals the remarkable performance achieved by both the Decision Tree and Extreme Gradient Boosting algorithms, attaining high accuracy rates of 99.9%. Following closely, k-Nearest Neighbor demonstrated commendable accuracy at 99.8%, while Random Forest exhibited a slightly diminished accuracy of 78.2%.

Table VI
SOME METRICS OF MULTI-CLASSIFICATION

| Evaluation | DT | RF | KNN | XGB |
|---|---|---|---|---|
| ACC | 99.9% | 78.2% | 99.8% | 99.9% |
| PRE | 99.9% | 59.6% | 99.7% | 100.0% |
| RE | 99.7% | 46.1% | 98.7% | 99.8% |
| F1 | 99.6% | 49.8% | 99.2% | 99.9% |
| Time (s) | 11.5 | 206.1 | 85.2 | 956.2 |

Precision (PRE), which assesses the ability to correctly classify instances within each class, underscores the exceptional precision scores attained by the Decision Tree (99.9%) and Extreme Gradient Boosting (100.0%) algorithms. k-Nearest Neighbor demonstrated a commendable precision of 99.7%, while Random Forest displayed a relatively lower precision of 59.6%.

The metric of recall (RE), which reflects the sensitivity to correctly classify instances within each class, showcased outstanding results for the Decision Tree (99.7%) and Extreme Gradient Boosting (99.8%) algorithms. k-Nearest Neighbor exhibited a recall rate of 98.7%, while Random Forest achieved a recall of 46.1%, indicating a relatively lower performance in this aspect.

The F1 score, serving as a comprehensive evaluation of both precision and recall, further highlights the remarkable performance of the Decision Tree (99.6%) and Extreme Gradient Boosting (99.9%) algorithms. k-Nearest Neighbor achieved a respectable F1 score of 99.2%, while Random Forest yielded a comparatively lower score of 49.8%.

In terms of computational time, the Decision Tree algorithm demonstrated the shortest processing time of 11.5 seconds, showcasing its efficiency. k-Nearest Neighbor required 85.2 seconds, while Random Forest consumed a longer processing time of 206.1 seconds. On the other hand, the Extreme Gradient Boosting algorithm exhibited the highest computational demand, with a processing time of 956.2 seconds.

The findings of this evaluation highlight the superior performance of the Decision Tree and Extreme Gradient Boosting algorithms across multiple metrics, including accuracy, precision, recall, and F1 score. Additionally, the Decision Tree algorithm stands out for its computational efficiency. However, it is important to consider the specific requirements and priorities of the application when selecting the most suitable algorithm, taking into account a balanced consideration of accuracy, precision, recall, and computational efficiency. The results of this research offer valuable insights into the performance characteristics of various machine learning algorithms for multi-classification in IoT-botnet problems.

## 3. Models evaluation

In order to ascertain the suitability of a given model for a particular problem, performance evaluation techniques are utilized. A range of methodologies may be employed, including but not limited to accuracy, precision, recall, and F1-score measurements. These methods serve as reliable indicators of a model's ability to effectively address the original problem at hand.

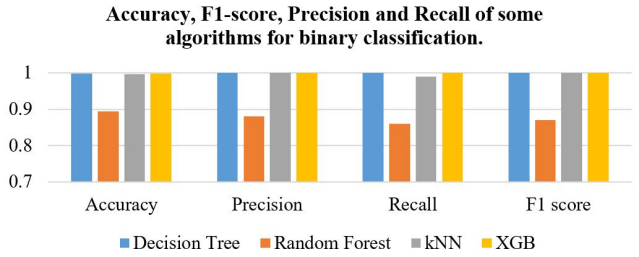



Figure 3. Accuracy, F1-score, Precision and Recall of some algorithms for binary classification

The performance evaluation of the DT, RF, KNN, and XGB algorithms for binary classification is presented in Figure 3, providing a comprehensive overview of the accuracy, precision, recall, and F1-score metrics. The results distinctly demonstrate the outstanding performance exhibited by both the DT and XGB algorithms, surpassing their respective counterparts. Additionally, in terms of training time, the Decision Tree algorithm emerges as the more efficient option, thereby delivering superior overall performance.

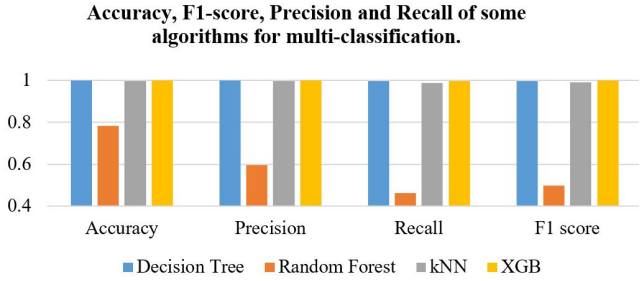



Figure 4. Accuracy, F1-score, Precision and Recall of some algorithms for multi-classification

The multiclass problem was evaluated to assess the performance of four distinct algorithms, namely Decision Tree, Random Forest, KNN, and XGB, as depicted in Figure 4. The evaluation encompassed the comprehensive analysis of accuracy, precision, recall, and F1-score metrics. Notably, the results demonstrate that the Decision Tree algorithm surpasses the other algorithms in terms of these performance metrics, showcasing superior performance.

Figure 5 presents the training time of the algorithms for the binary classification problem. The findings reveal that the XGB algorithm exhibits significantly longer training time compared to the other algorithms. Conversely, the decision tree algorithm emerges as the most efficient in terms of training time, outperforming all other algorithms in this regard.

In the multi-classification problem, the training time of the algorithms is presented in figure 6. The results illustrate


Figure 5. Training time for binary classification

that the XGB algorithm takes significantly longer to train compared to the other algorithms, with a training time almost 100 times that of the decision tree algorithm. On the other hand, the decision tree algorithm exhibits the best performance in terms of training time.

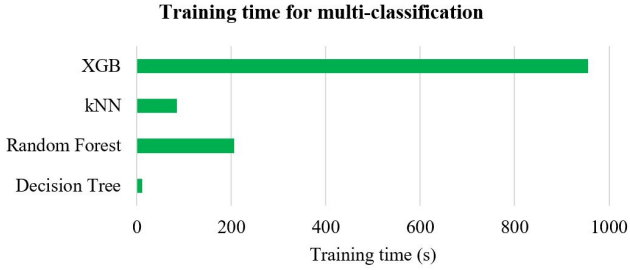



Figure 6. Training time for multi-classification

## V. CONCLUSIONS

In conclusion, this scholarly research paper has undertaken the critical task of detecting and categorizing IoT botnets using machine learning algorithms. The study has placed considerable emphasis on the thorough analysis and manipulation of IoT botnet data, with a specific focus on the highly regarded IoT-23 dataset. By deploying widely recognized and extensively utilized machine learning algorithms such as Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), the objective was to effectively classify and detect botnets within the confines of the IoT-23 dataset. The application of these algorithms has aimed to advance our understanding of their performance and effectiveness in the realm of IoT botnet detection and classification tasks.

Through a comparative analysis of the outcomes derived from these diverse algorithms, valuable insights have been acquired, shedding light on their respective strengths and limitations. Such insights equip researchers and practitioners with the knowledge required to make well-informed decisions when choosing the most appropriate algorithm for successful IoT botnet detection and classification. This research contributes significantly to the field by providing a comprehensive evaluation of the performance of these machine learning algorithms, thereby facilitating the development of more resilient and efficient detection and classification methodologies.

By addressing the pivotal challenge of IoT botnet detection and classification, this study has implications that extend to the realm of enhancing cybersecurity measures within IoT ecosystems. The findings and insights gleaned from this research have the potential to inform the development of proactive measures aimed at preventing and mitigating the detrimental effects of IoT botnet attacks. It is anticipated that this work will inspire further research endeavors and foster collaboration within the field, ultimately leading to the advancement of more secure IoT systems and the safeguarding of critical infrastructures.

This research study opens up promising avenues for future exploration and advancement in the realm of IoT botnet detection and classification. Firstly, there is a need to undertake comprehensive exploration and evaluation of additional machine learning algorithms to expand the repertoire of options available for effectively detecting and categorizing IoT botnets. Incorporating newer algorithms and techniques holds the potential to enhance the overall performance and accuracy of the detection process. Additionally, the exploration of ensemble learning approaches, where multiple algorithms are combined synergistically, presents an intriguing avenue to potentially unlock improved results and a more robust detection framework.

## REFERENCES

[1] Williams, P., Dutta, I.K., Daoud,. H, and Bayoumi, M, *A survey on security in internet of things with a focus on the impact of emerging technologies*, Elsevier. (2022).

[2] Khan, W.Z., Rehman, M.H., Zangoti, H.M., Afzal, M.K., Armi, N., and Salah, K, *Industrial internet of things: Recent advances, enabling technologies and open challenges*, Elsevier, 2020..

[3] Vitorino, J., Andrade, R., Praça, I., Sousa, O., and Maia, E, "A Comparative Analysis of Ma-chine Learning Techniques for IoT Intrusion Detection", *Foundations and Practice of Security* (pp. 191-207), Springer, (2022).

[4] Haq, N.F., Onik, A.R., Hridoy, M.A.K, Rafni, M., Shah, F.M., and Farid, D.M, "Application of Machine Learning Approaches inIntrusion Detection System: A Survey", *International Journal of Advanced Research in Artificial Intelligence(IJARAI)*, Volume 4 Issue 3, (2015).

[5] Hajji, J., Khalily, M., Moustafa, N., and Nelms, T. IoT-23, *A Dataset for IoT Network Traffic Analysis*, Springer, (2019).

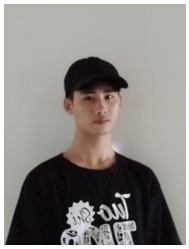
[6] Rahim, A., Razzaque, M.A., Hasan, R., and Hossain, M.F, *Effective IoT Network Security through Feature Selection and Machine Learning Techniques*, IEEE, 2020..

[7] Islam, S.M.Z, Bhuiyan M.Z.H, and Hasan R., *Fusion of Machine Learning Models for Intru-sion Detection in IoT Networks using the IoT-23 Dataset*, IEEE, 2020.

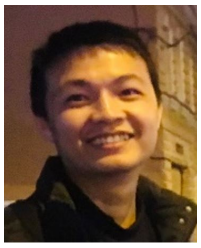
[8] Li, Y., Qiu, L., Chen, Y., and Chen, Y., *Ensemble-based Intrusion Detection System for IoT Networks using the IoT-23 Dataset*, IEEE, 2020.

[9] P. H. Do, T. D. Dinh, D. T. Le, V. D. Pham, L. Myrova and R. Kirichek, "An Efficient Fea-ture Extraction Method for Attack Classification in IoT Networks," *13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2021.
[10] Alotaibi, F., Al-Qaness, M.A., Abunadi, A., and Alghazzawi, M.A., *A Deep Learning Ap-proach for Intrusion Detection in IoT Networks using the IoT-23 Dataset*, IEEE, 2020.
[11] Li, J., Hu, C., Yang, K., Zhang, X., and Lu, J, *An IoT-23 based IoT Intrusion Detection Sys-tem using Deep Learning*, IEEE, 2020..
[12] Abdallah, A., Khalil, I., Al-Emadi, N., Almohaimeed, A., and Kim, H., *Real-Time IoT Botnet Detection Using Deep Learning on IoT-23 Dataset*, IEEE, 2020.
[13] Kiani, A.T., Abbas, R.A., Abbasi, A.Z., and Khan, M.K., *Deep Learning-based Anomaly De-tection for IoT Networks using the IoT-23 Dataset*, IEEE, 2020.
[14] Rasool, S., Saeed, S., Farooq, F., and Madani, A., *A Comparative Study of Transfer Learning Approaches for IoT Malware Detection Using IoT-23 Dataset*, IEEE, 2021.
[15] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga, *IoT-23: A labeled da-taset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]*, Zenodo, 2020.
[16] Stoian, N.A., "Machine Learning for Anomaly Detection in IoT Networks : Malware analysis on the IoT-23 data set", *EEMCS: Electrical Engineering, Mathematics and Computer Science*, 2020.

**Pham Van Quan** is currently a senior student majoring in Data Science and Artificial Intelligence in Dong A University. His research interests include data science, machine learning, AI and its application in different fields like Finance, Network and Natural Language Processing.
Email: quan10.work@gmail.com.



**Ngo Van uc** became a student at Dong A University in 2020, majoring in Artificial Intelligence and Data Science. His research interests include machine learning, deep learning, data science, artificial intelligence, image processing, and their applications.
Email: ngovanuc.1508@gmail.com.



**Do Phuc Hao** received his MS degree in Computer science from the University of Danang - University of Science and Technology in 2017. He is currently a Ph.D. student in the Department of Communication Networks and Data Transmission at the Bonch-Bruevich Saint- Petersburg State University of Telecommunications, Russia. His research interests include Artificial Intelligence, Machine Learning and its application in different fields like network, blockchain.
Email: haodp@dau.edu.vn.

**Nguyen Nang Hung Van** received his Ph.D. degree in Computer Science from The University of Danang, Vietnam, in 2021. He is currently a lecturer at The University of Danang - University of Science and Technology. His research interests include Artificial Intelligence, Machine Learning, Geometric Algebra, Computer Networking.
Email: nguyenvan@dut.udn.vn.