JOURNAL

JOURNAL OF RESEARCH AND DEVELOPMENT ON INFORMATION AND COMMUNICATION TECHNOLOGY

ISSN 1859-3534

TẠP CHÍ NGHIÊN CỨU VÀ PHÁT TRIỂN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG - MINISTRY OF INFORMATION AND COMMUNICATIONS

Volume 2020

Number 2 December 2020

Chuyên san
Các công trình nghiên cứu phát triển công nghệ thông tin và truyền thông

Journal Address
115 Tran Duy Hung Rd., Cau Giay District, Hanoi
Tel: (84)-24-37737136, Fax: (84)-24-37737130,
Website https://ictmag.vn
Email: chuyensanbcvt@mic.gov.vn
Journal Office Branch
27 Nguyen Binh Khiem Street, District 1, Ho Chi Minh City, Vietnam
Tel/Fax: (84)-28-38992898

Journal marketing and distribution
Pham Thi Lam
Email: ptlam@mic.gov.vn, Tel: 84-948503999

License No 365/GP-BTTTT, dated 19th December 2014;
Modified license No 233/GP-BTTTT dated 23rd May 2017.
Printed by Cong ty TNHH MTV In Tap chi Cong san.
Printed and delivered in January 2021.

Price: 40,000 VND

Research and Development on
Information and Communication Technology
https://ictmag.vn
Volume 2020, Number 2
December

TABLE OF CONTENTS

REGULAR ARTICLES

2020 LIST OF REVIEWERS

Ishaani Priyadarshani	University of Delaware. USA
Nguyễn Quang Uy	Le Quy Don University of Technology, Viet NamM
Giuseppe D'Aniello	University of Salerno, Italia
Nguyễn Phi Lê	Hanoi University of Science and Technology
Nguyễn Hữu Phát	Hanoi University of Science and Technology
Trần Đức Tân	VNU Hanoi University of Engineering and Technology
Phạm Văn Hải	Hanoi University of Science and Technology
Hieu Nguyen	Case Western Reserve University, USA
Mai Đình Sinh	Le Quy Don University of Technology, Viet Nam
Trần Nguyên Ngọc	Hanoi University of Science and Technology
Nguyễn Hoài Sơn	Hochiminh University of Technology and Education
Từ Minh Phương	Posts and Telecommunications Institute of Technology
Trần Trung Duy	Posts and Telecommunications Institute of Technology
Lâm Sinh Công	VNU Hanoi, University of Engineering and Technology
Nguyễn Tuấn Đăng	Sai Gon University
Nguyễn Thanh Hiên	Bank University Hochiminh
Lê Hoàng Thái	VNU Hochiminh, University of Science
Phạm Thế Bảo	Sai Gon University
Nguyễn Đức Dũng	Vietnam Academy of Sicience and Technology, Institute of Information Technology
Nguyễn Tài Hưng	Hanoi University of Science and Technology
Nguyễn Ngọc Hóa	VNU Hanoi, University of Engineering and Technology
Nguyễn Mạnh Hùng	Posts and Telecommunications Institute of Technology
Đỗ Thị Bích Ngọc	Posts and Telecommunications Institute of Technology

CAM-D: A Description Method for Multi-cloud Marketplace Application

Hoang-Long Huynh[1], Duc-Huu Nguyen[1], Trong-Vinh Le[2], Quyet-Thang Huynh[1]
[1] Hanoi University of Science and Technology, Hanoi, Vietnam
[2] University of Science, Vietnam National University, Hanoi, Vietnam
Correspondence: Hoang-Long Huynh, longlove1232002@yahoo.com
Communication: received 30 November 2020, revised 27 January 2021, accepted 31 January 2021
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n2.943

***Abstract*: Multi-cloud Marketplace facilitates to create a diverse ecosystem for cloud software and cloud resource services provided by many stakeholders. To leverage the advantage of multi-cloud environment, cloud application could be a composition of software components which are able to be distributed across various cloud providers. So the cloud application is therefore a complex system. Consequently, an important key problem remained in research is to define multi-cloud application in a particular form and to construct his description. In this paper, we particularly focus on developing a description method that can be taken to tackle the lack of description for multi-cloud application by designing description templates for CAM which was developed in [1][2][3], called CAM-D. A completed application description can be synthesized from individual component descriptions. Our experimentation is expressed through the transformation of CAM-D template to TOSCA specification illustrated by case study. In addition, we also develop an flattening algorithm to assist in mapping to TOSCA.**

Keywords: *Cloud Computing, Multi-cloud, Multi-cloud Marketplace, Multi-cloud Marketplace Application, Composable Application Model (CAM), Software as a Service, Description Method.*

I. INTRODUCTION

At present, Cloud marketplace is an effective method for cloud providers to delivery Software as Service (SaaS) in a convenient way to consumers. Nevertheless, almost existing cloud marketplaces are owned by vendors themselves, and SaaS development is tightly coupled to Application Programming Interfaces (APIs) and tools of individual cloud providers. To overcome vendor lock-in problem, our ideal is about a new Multi-cloud marketplace model, called O-Marketplace [4], his ecosystem is depicted in Figure 2 including four actors: Cloud Consumers, Cloud App Vendors/Developers, Cloud Providers and O-Marketplace Administrator. SaaS supplied by O-Marketplace should be tailored to meet the characteristics of multi-cloud by separating cloud application into two parts, i.e. *cloud software* and *cloud platform* (runtime system), and cloud software could designed beyond a certain cloud provider. Ideally, cloud software is decomposed into components which are independently developed by various developers and can be deployed across different clouds. The overall idea of Multi-cloud application is illustrated in Figure 1 and shaped in our previous studies [1][2], called Composable Application Model (CAM).

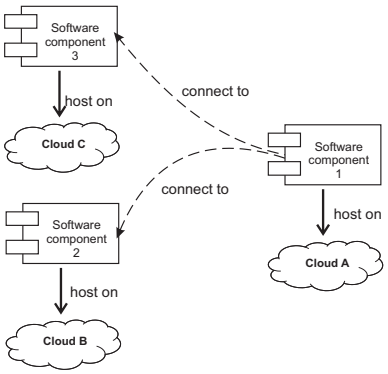

Figure 1. Multi-Cloud Application

In another aspect, multi-cloud marketplace has required manageability of cloud application. As such, there is an urgent need to have a written-related multi-cloud application description that predate, or have been developed. Unfortunately, there are very few existing related works to resolve this issue. There are three main approaches to describe multi-cloud application. The first one is a much-mentioned approach which is to use Domain Specific Language (DSL) to describe cloud application presented by K. Sledziewski et al. [5], E. Brandtzaeg et al. [6], G. Baryannis [7], G. Sousa [8], and N. Ferry [9]. The second

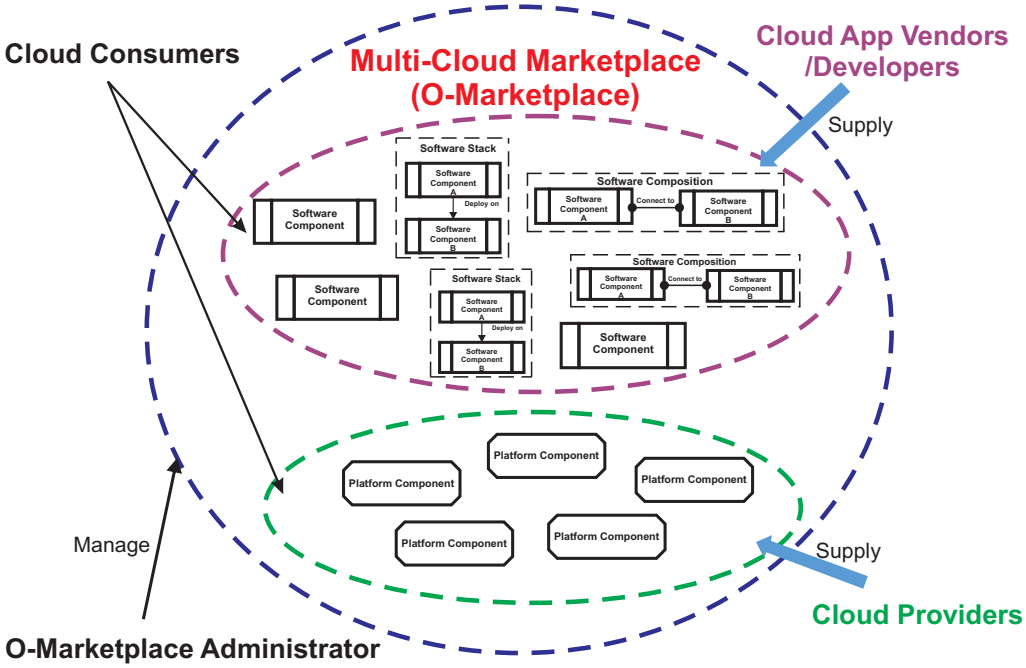

Figure 2. O-Marketplace Ecosystem

approach is to use open source solutions such as Heat [10] and Juju [11] for describing and modeling composite cloud applications and supporting deploying them on particular cloud providers. The last approach is to adopt TOSCA specification to distribute a cloud application across multiple clouds introduced by G. Tricomi [12], J. Carrasco [13], K. Alexander and [14], and K. Saatkamp [15].

In general, two first approaches could be done on a single cloud, and despite two notable efforts refined TOSCA [16] to enable a customized distribution of the components of an application to different cloud providers introduced in [13][15] which mentioned in the last, these proposals have limitation as follows: (i) there is no proposal to independently describe components within a multi-cloud application; (ii) the lack of solutions that are capable of combining individual component descriptions to form a complete description of a multi-cloud application. (iii) the multi-cloud application specification validation mechanism has not been built in yet based on proposed specification/description methods.

To tackle these limitations, our key approach is to develop a description method which is able to cover multi-cloud application in a uniform. This supports the distribution of software components on heterogeneous multi-cloud platforms. In this paper, we define a specification method for multi-cloud application which is modeled by CAM introduced in previous papers [4][1][2][3]. Our work towards the following contributions:

- Defining the description templates for CAM.
- Presenting a novel approach for creating the multi-cloud application description: the completed description of multi-cloud application is synthesized from the individual component descriptions.
- For experimentation, we propose an algorithm for flattening the description to TOSCA-based specification.

The rest of the paper is organized as follows: the core of our work in this paper is presented in Section II, we focus on making up description templates for CAM. In Section III, the illustration of transformation from CAM-D description template to TOSCA specification is shown. The advantages of CAM-D is highlighted in Section V. Finally, we conclude and summarize the contributions of the paper in Section VI.

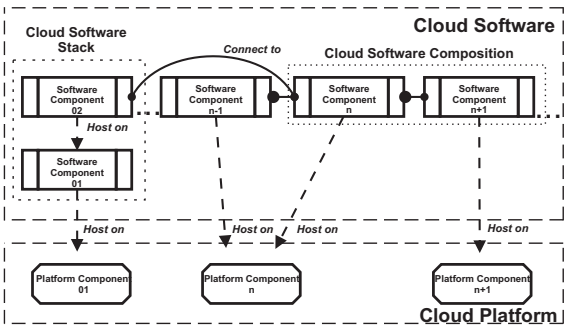

Figure 3. Composable Application Model [2].

II. CAM-based Description Templates

With the motivation to standardize the multi-cloud application description in a uniform, we propose a description method that is not only capable of expressing its behaviours and properties in standardized description patterns, but also meet the multi-cloud characteristics in general and O-Marketplace proposed in [4] in particular. To achieve this goal, our work focuses on developing description templates for entire CAM and its components, which were defined in our recent paper [2] including: Simple Cloud Software Component, Cloud Software Stack, Cloud Software Composition, Cloud Platform Component, and CAM depicted in Figure 3. The description of CAM application can be built from individual descriptions based on the *matching rules* proposed in [2]. In addition, an interesting point is that CAM is defined as a nested structure, so the description method has to express this special feature.

1. CAM-based Application

Before going further into details of the CAM description templates, several conventions have been made as follows:

- All templates are written in a YAML-like format.
- We assume that all the defined templates are available so that they can be referred via their ids.
- Since CAM components are designed in a nested manner, we use dot-notation for accessing to an inner element and/or property of a component. For example, `c.capabilities[cid].p` stands for the property `p` of the capability `cid` of the component `c`.

```
application <app id> =
  software : <software component template>
    arguments:
      - <arg> : <value>
      - ...
  platforms :
    - <platform> : <platform service
        template>
      arguments:
        - <arg> : <value>
        - ....
    - ...
  dependencies:
    - <dependency>:
      type: HOST-ON
      from: <software>.<requirement>
      to: <platform>.<capability>
end
```

Figure 4. CAM-based Application Template

A CAM-based multi-cloud application can be described by an application script using the application template depicted in the Figure 4. Each application should specify a CAM-based cloud software component as well as a cloud platforms on which we prefer to host the application. The cloud software component and the platform are characterized by sets of parameters which serve as user preferences. The application script should also indicate the correspondences between the software requirements and the platforms in the section `dependencies`.

2. Platform Service Template

Platform services (*Cloud Platform Component* depicted in Figure 5) are provided by cloud providers. In order to integrate their services into our CAM-based system, cloud providers should publish their services in form of Platform Services Template as shown in the Figure 6.

Figure 5. Cloud Platform Component [2].

```
platform <platform id> =
  capabilities:
    - <capability> : <port signature>
    - ...
  parameters:
    - <param> : <type>
    - ...
  parameter mappings:
    - <capability>.<property> as <param>
        default <value>
    - ...
  methods :
    - <method> : <method type>
      parameters:
        - <arg> : <type>
        - ....
    - ...
end
```

Figure 6. CAM-based Platform Service Template

Each cloud platform service defines a set of capabilities to provide to the cloud software. These capabilities should match with the requirements of the software for the successful deployment of the application. In order to perform the matching algorithm, each capability and/or requirement is associated to a port signature (illustrated in Figure 7) which describes the properties and the operations provided by the capability and/or needed for the requirement. Please note that the a port signature is not only specifies the dependency between a cloud software requirement and the platform capability, but also is used to describe the dependency

among software components in a composition. We distinguish different kinds of dependency on a port signature by port type which is either HOST-ON or CONNECT-TO.

```
signature <port signature id> =
  type: <port type>
  properties:
    - <property> : <type>
    - ...
  methods :
    - <method> : <method type>
      parameters:
        - <arg> : <type>
        - ....
    - ...
end
```

Figure 7. Port Signature

3. General Structure of a Cloud Software Component Template

CAM-based cloud softwares and components are defined in a uniform way using the Cloud Software Component template (Figure 8).

Each component may have its own inner structure which consists of a set of sub-components and the dependencies (described in the Section `Dependencies` of the template) among them. A CAM runtime should check the well-form of a component by matching requirements to capacities according to the specified dependencies. In a special case, when the component is a Simple Software Component, this inner structure is skipped.

The outer view of a CAM-based component consists of a set of capabilities (described in the Section `Capabilities` of the template) specifying what the component provides, and a set of requirements (described in the Section `Requirements` of the template) specifying what are needed for managing and running the component.

Properties of these capabilities are mapped into a set of parameters in the section `parameter mapping` so that programmers will know how to characterize the components when they are in use. In some special cases, the component itself has extra parameters defined in the section `parameters`. This is worth to mention that the properties of sub-components requirements can be taken from the properties of their corresponding capabilities from platforms and/or other sub-components.

A component also provides a set of operations indicated by methods in the `methods` section. The component script should describes the set of arguments to a method as well as its execution code, normally written in a special scripting language.

```
component <component id> =
  type: <component type>
  # Inner structure of the component
  sub-components:
    - <component> : <software component
        template>
    - ...
  dependencies:
    - <dependency>:
      type: <port type>
      from: <component>.<requirement>
      to: <component>.<capability>
    - ...
  # outer interface
  capabilities:
    - <capability> map-to
        <component>.<capability>
    - <capability> : <port signature>
    - ...
  requirements:
    - <requirement> map-to
        <component>.<requirement>
    - <requirement> : <port signature>
    - ...
  parameters:
    - <param> : <type>
    - ...
  parameter mappings:
    - <capability>.<property> as <param>
        default <value>
    - ...
  methods :
    - <method> : <method type>
      parameters:
        - <arg> : <type>
        - ...
      code: <script>
    - ...
end
```

Figure 8. General Structure of a Cloud Software Component Template

4. Simple Software Component Template

Simple Cloud Software Component (SSC) is the atomic entity or the basic element of CAM. A SSC packs its code and data together with the necessary requirements and capabilities for deployment and operation. The model of SSC is depicted in Figure 9. SSC can only reside in a single cloud platform.

As mentioned before, a SSC doesn't have its own inner structure. Instead, programmers should define component's capabilities and requirements by specifying corresponding port signatures. All of the code and data to manage and run the component can be defined in the `methods` section.

A Simple Software Component can therefore be defined using the template shown in Figure 10.

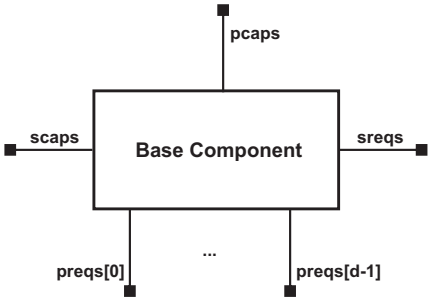

Figure 9. Simple Cloud Software Component [2].

```
component <component id> =
  type: SimpleComponent
  capabilities:
    - <capability> : <port signature>
    - ...
  requirements:
    - <requirement> : <port signature>
    - ...
  parameters: ...
  parameter mappings: ...
  methods: ...
end
```

Figure 10. Simple Software Component Template

5. Cloud Software Stack Template

Cloud Software Stack (SS) represents a series of software components established on each others deployed on a single cloud platform. For simplicity, we define a SS with two elements: a top component and a base component. In the nested manner, we restrict top component to be a simple component while base component can be either a simple component or another stack.

Composing elements of SS depicted in Figure 11 and we define the the description template for SS as showed in Figure 12.

A description template of SS is well-formed if it satisfies the following validation conditions:

- Top component and base component should be well-formed;
- Type of the top component should be “SimpleComponent";
- Type of the base component should matches either “SimpleComponent" or “SoftwareStack";
- The dependencies between the top component and the base component defined in the section `dependencies` should be guaranteed, i.e. signatures of the top component requirements should match with the signatures of the base component capabilities;

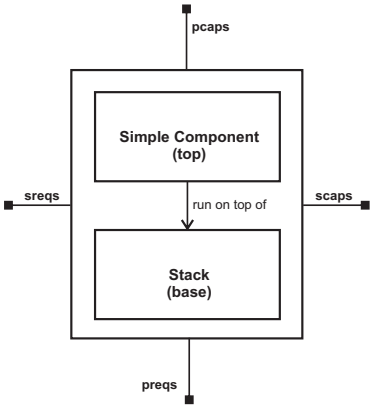

Figure 11. Cloud Software Stack [2].

```
component <component id> =
  type: SoftwareStack
  # inner structure
  sub-components:
    - top: <Component template>
    - base: <component template>
  dependencies:
    - <dependency>:
      type: HOSTON
      from: top.requirements[<requirement>]
      to: base.capabilities[<capability>]
    - ...
  # outer interface
  capabilities: ...
  requirements:
  parameters: ...
  parameter mappings: ...
  methods: ...
end
```

Figure 12. Software Stack Component Template

6. Cloud Software Composition Template

Cloud Software Composition (SC) represents a distributed multi-cloud software composition including cloud components which are able to deploy on various cloud platforms. Each component may connect to others via software protocols. In our CAM model, we defined such software protocols as port signatures of type “CONNECTTO".

For simplicity, we define a SC with 2 components: left and right. Composing elements of SC is depicted in Figure 13, and description template of Cloud Software Composition is shown in Figure 14.

The description template of a SC is well-formed if it satisfies the following validation conditions:

- All of its sub-components (i.e. left and right) are well-formed.

- The dependencies between the left component and the right component defined in the section `dependencies` should be guaranteed, i.e. signatures of the left component requirements should match with the signatures of the right component capabilities;

Please note that we can extend the template for software composition consisting more than two sub-components. In this case, a dependency between sub-components is a mapping from requirement of a sub-component to a capability of its counter part. This extended template will be used for describing *flattening composition* representing in the next section.

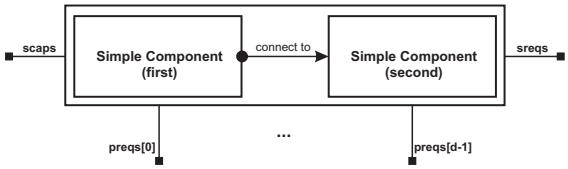

Figure 13. Cloud Software Composition [2].

```
component <component id> =
  type: SoftwareComposition
  # inner structure
  sub-components:
    - left: <Component template>
    - right: <component template>
  dependencies:
    - <dependency>:
      type: CONNECT-TO
      from: left.requirements[<requirement>]
      to: right.capabilities[<capability>]
    - ...
  # outer interface
  capabilities: ...
  requirements:
  parameters: ...
  parameter mappings: ...
  methods: ...
end
```

Figure 14. Software Component Composition Template

III. Transforming from CAM-D to TOSCA specification

In order to prove the feasibility of our approach, it is necessary to have a runtime system which is able to parse our CAM-D application script and deploy the application on specified cloud platforms. While this is still on our on-going research, in this paper, we employ an alternative method: translating CAM-D application script to well-known TOSCA cloud application model and then using existing TOSCA-based runtime for deploying the application.

In general, our CAM-D application description template is very similar to TOSCA topology template. In terms of topology, the concept of Node in TOSCA template is equivalent to our CAM-D software component, and the concept of Relationship in TOSCA is equivalent to our CAM-D dependency. Different from TOSCA, CAM-D represents a cloud application in a nested manner for which software components can be independently developed and composed by different developers.

To transform a CAM-D application script to TOSCA topology template, we apply a two-step process:

1) Flattening the given CAM-D application script into a *flattening composition*;
2) Transforming the *flattening composition* into TOSCA topology template

In the following sub-sections, we will give a short representation about TOSCA as well as our two steps of the transformation algorithm.

1. Topology and Orchestration Specification for Cloud Application

Topology and Orchestration Specification for Cloud Applications (TOSCA) [16] is an open standard built by OASIS that defines the inter-operable description of cloud application hosted on the cloud. TOSCA will enable the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behavior of these services independent of the supplier creating the service, and any particular cloud provider (Figure 15).

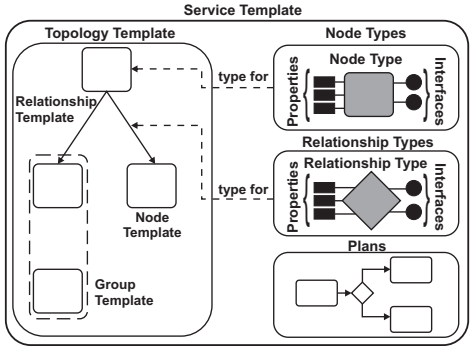

Figure 15. TOSCA Service Template overview [16].

TOSCA will enable portable deployment to any compliant cloud and enhance the portability multi-cloud provider applications. So TOSCA is also very efficient in software provisioning, deployment and management of cloud application. It has been supported by many partners like IBM, Reb Hat, Cisco, Citrix, EMC, etc. Thus, we utilize this

standard as a target of the transformation to demonstrate the feasibility of our proposed description method.

2. Flattening Algorithm

The first step of the transformation algorithm from CAM-D application script to TOSCA topology template is to flatten the nested composition of the CAM-D software component into a flat composition where all sub-components of the flat composition should be of type SimpleComponent. The transformation should keep all external view of the component, i.e. all requirements, capabilities, parameters, and methods. The things should be change are internal structure of the composition as well as the mapping of outer requirements, capabilities, parameters to its corresponding inner elements.

This would be done easily by using a recursive algorithm shown as below:

```
00. Flattening Algorithm (CAM-D script);
01. Input:
02.   - A nested software component
03. Output:
04.   - A flattening software component
05.   - A mappings from capabilities / requirements / parameters of input sub-components
06.     to the correspondences of the output simple sub-components.
07. Begin
08.   If c.type = "SimpleComponent"
09.   Then
10.       M = {e->c.e, for each capability / requirement / parameter e of c}
11.       Return (c,M)
12.   Else If c.type = "SoftwareStack"
13.   Then
14.    Begin
15.       (new-top, top-mappings) = Flatten(c.top)
16.       (new-base, base-mapping) = Flatten(c.base)
17.       mappings = AddPrefix("top", top-mapping) ⋃
18.                  AddPrefix("base", base-mappings)
19.       sub-components = new-top.sub-components ⋃ new-base.sub-components
20.       dependencies = new-top.dependencies ⋃ new-base.dependencies ⋃
21.                      UpdateDependencies(c.dependencies, mappings)
22.       requirements = UpdateRequirements(c.requirements, mappings)
23.       capabilities = UpdateCapabilities(c. capabilities , mappings)
24.       parameter-mappings = UpdateParameters(c.parameter-mappings, mappings)
25.       methods = UpdateMethods(c.methods, mappings)
26.       new-mappings = MakeMappings(requirements, capabilities, parameter-mappings)
27.       Return (FlatCompositions(sub-components, dependencies,
28.                  requirements, capabilities , c.parameters, parameter-mappings, methods),
29.               new-mappings)
30.    End
31.   Else If c.type = "SoftwareComposition"
32.  Then
33.    Begin
34.       (new-left, left -mappings) = Flatten(c. left )
35.       (new-right, right -mapping) = Flatten(c. right )
36.       mappings = AddPrefix("left", left -mapping) ⋃
37.                  AddPrefix("right", right -mappings)
38.       sub-components = new-left.sub-components ⋃ new-right.sub-components
39.       dependencies = new-left.dependencies ⋃ new-right.dependencies ⋃
40.                      UpdateDependencies(c.dependencies, mappings)
41.       requirements = UpdateRequirements(c.requirements, mappings)
42.       capabilities = UpdateCapabilities(c. capabilities , mappings)
43.       parameter-mappings = UpdateParameters(c.parameter-mappings, mappings)
44.       methods = UpdateMethods(c.methods, mappings)
45.       new-mappings = MakeMappings(requirements, capabilities, parameter-mappings)
46.       Return (FlatCompositions(sub-components, dependencies,
47.                  requirements, capabilities , c.parameters, parameter-mappings, methods),
48.               new-mappings)
49.    End
50. End;
```

In this algorithm, the function `AddPrefix(s, mappings)` add a component access prefix `s` to all origins of `mappings` to avoid ambiguity. The functions `UpdateDependencies`, `UpdateRequirements`, `UpdateCapabilities`, `UpdateParameters`, `UpdateMethods` are just used for updating all references to the requirements/capabilities/parameters of sub-components of the input component by the corresponding elements of the SimpleComponent sub-components in the target component. The function `MakeMappings` just accumulates all modified mappings of requirements, capabilities, and parameters. Due to the limitation of the paper, we omit the presentation of these functions.

Correctness: The flattening algorithm is correct in the following principles:

- All sub-components of a resulting flat component are components of type SimpleComponent;
- Dependencies between the top and the base sub-components of any SofwareStack inside the structure of input component should be reflected in the dependencies of resulting flat component.
- Dependencies between the left and the right sub-components of any SofwareComposition inside the structure of input component should be reflected in the dependencies of resulting flat component.
- All mappings (requirement mappings, capability mappings, parameter mappings) of the input component should be changed in the output flat component so that they maps their original external names to specific names of requirements/capabilities/parameters of corresponding SimpleComponents.

The correctness of the algorithm can be easily proved by reduction on the hierarchical structure of the input nested component.

Complexity: The flattening algorithm is proceeded recursively on the hierarchical structure of the input nested component. Thus the complexity of the algorithm is linear to the number of the sub-components inside that structure, i.e. O(n) for n is the number of sub-components.

3. Mapping to TOSCA

As a result of the above flattening algorithm, a CAM-D application will be transformed into a flat model in which all sub-components are of type SimpleComponent.

The second step of the transformation algorithm is to translate this flat application model together with its undelying platforms into TOSCA topology template. The translations is rather simple. Each sub-components and/or platforms will be transformed into a TOSCA node. Each dependency among components and between a component and a platform will be translated into a TOSCA relationship. We do not give further details of this translation algorithm. Instead, result from an experimentation which will be given in the next section will demonstrate our mapping method.

IV. Experimentation with Case Study

In this section, we demonstrate the feasibility of CAM-D by translating flattening description template of CAM-D into TOSCA specification template with a case study.

1. Case Study

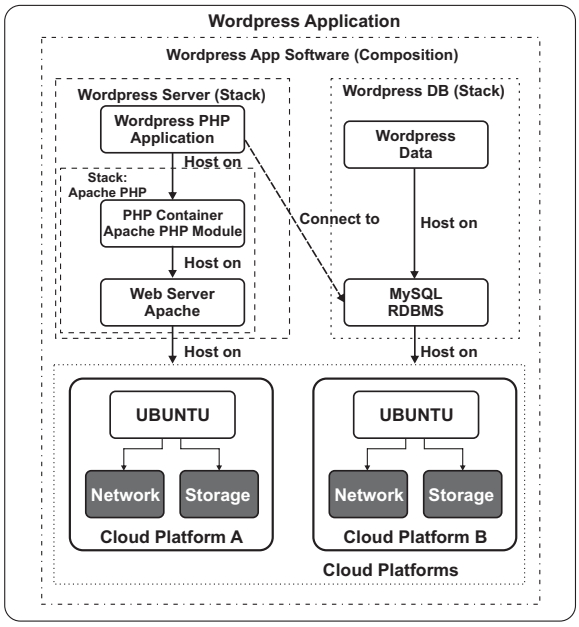

Figure 16. Wordpress Application modeled by CAM

```
application Wordpress-Application =
   software: Wordpress-Software
      arguments:

   platforms :
         p1 : VM-Platform
            arguments:
               provider: "dsg@openstack"
               instanceType : "000001920"
               baseImage :
                   "a82e054f-4f01-49f9-bc4c
               -77a98045739c"
         p2 : VM-Platform
            arguments:
               provider: "dsg@flexiant"
               instanceType : "000001920"
               baseImage :
                   "a82e054f-4f01-49f9-bc4c
               -77a98045739c"
   dependencies:
      d1:
         type: HOST-ON
         from: software.requirements[R1]
         to:  p1.capabilities[vm]
      d2:
         type: HOST-ON
         from: software.requirements[R2]
         to:  p2.capabilities[vm]
end
```

Figure 17. Wordpress Application Description Template

For validate our idea, we deploy a WordPress application on two Clouds: OpenStack and Flexiant. We use CAM to model Wordpress application which is depicted in Figure 16. There are five software components of cloud software: Wordpress PHP Application, PHP Container, Apache, Wordpress DB, MySQL covered in two stack, and two Platform components are cloud infrastructures: OpenStack and Flexiant. These software components modeled in three software stack and one software composition. The CAM-D template of Wordpress Application showed in Figure 17.

According CAM-D, the description template of Wordpress Application is created based on individual description templates which represent for the type of CAM-based components according to the degree of CAM. The overview of individual description templates of Wordpress Application is shown in Figure 18. The full code of Wordpress Application Templates is available at https://github.com/longlovehl/Descriptions/.

```
#########Wordpress Application#########
###Signature###
#Platform signature
signature VM-SIG =
end
#Apache signature
signature ApacheServer-SIG =
end
#PHPContainer signature
signature PHPContainer-SIG =
end
#MySQL-RDBMS signature
signature MySQL-CONNECTION-SIG =
end
signature MySQL-DEPLOYMENT-SIG =
end
###Description Template###
##Platform Template##
platform VM-Platform =
end
#Software Component Template
component ApacheServer =
end
component PHPContainer =
end
component Apache-PHP =
end
component Wordpress-PHP:
end
##Wordpress-server
component Wordpress-Server =
end
#Wordpress-DB--------
component MySQL-RDBMS =
end
component Wordpress-Data =
end
##Wordpress-DB
component Wordpress-DB =
end
##Wordpress-Software
component Wordpress-Software =
end
```

Figure 18. Wordpress Application Description Templates

2. Transforming CAM-D Template to TOSCA Specification

To transform CAM-D Template to TOSCA specification, we first use proposed flattening algorithm to flatten the Wordpress Application into a graph representation. Then, we map flattening description template of WordPress Application into TOSCA-based specification. All nodes of the graph are mapped to Node Template of TOSCA, implementation details of the Node Template are omitted from the specification for brevity. Edges of the graph are mapped to Relationship Template. According to the original of the edges, i.e., from a stack or from a composition, the type of corresponding Relationship template will be given as HOSTON or CONNECTTO. The result is showed in Figure 19.

```
<?xml version="1.0"?>
<ns2:Definitions id="WordpressApp"
      xmlns:ns2="http://docs.oasis-open.org/toscans/2011/12"
      name="WordpressApp">
<ns2:ServiceTemplate id="WordpressTopology">
 <ns2:TopologyTemplate>
   <ns2:RelationshipTemplate id="WordpressPHP_HostOn_PHPContainer"
      type="HOSTON">
    <ns2:SourceElement ref="PHPContainer"/>
    <ns2:TargetElement ref="WordpressPHP"/>
   </ns2:RelationshipTemplate>
   ...
   <ns2:RelationshipTemplate id="WordpressPHP_ConnecTTo_MySQLRDBMS"
      type="CONNECTTO">
    <ns2:SourceElement ref="MySQLRDBMS"/>
    <ns2:TargetElement ref="WordpressPHP"/>
   </ns2:RelationshipTemplate>
   <ns2:NodeTemplate id="ApacheVM" type="os" maxInstances="3" minInstances="1">
     <ns2:Properties>
      <MappingProperties>
       <MappingProperty type="os">
            <property name="provider">dsg@openstack</property>
            <property name="instanceType">000001920</property>
<property name="baseImage">a82e054f-4f01-49f9-bc4c-77a98045739c</property>
<property name="packages"/>
       </MappingProperty>
       </MappingProperties>
   ...
<ns2:NodeTemplate id="MySQLVM" type="os" maxInstances="2" minInstances="1">
      <ns2:Properties>
       <MappingProperties>
<MappingProperty type="os">
<property name="provider">dsg@flexiant</property>
<property name="instanceType">000001920</property>
<property name="baseImage">a82e054f-4f01-49f9-bc4c-77a98045739c</property>
<property name="packages"/>
</MappingProperty>
</MappingProperties>
      ...
    </ns2:NodeTemplate>
    <ns2:NodeTemplate id="MySQLRDBMS" type="software" maxInstances="1"
      minInstances="1">
      ...
      <ns2:Requirements>
       <ns2:Requirement id="MySQLVM" type="HOSTON"/>
      </ns2:Requirements>
      <ns2:Capabilities>
       <ns2:Capability id="WordpressPHP" type="CONNECTTO"/>
      </ns2:Capabilities>
      ...
 </ns2:NodeTemplate>
</ns2:TopologyTemplate>
</ns2:ServiceTemplate>
<ns2:ArtifactTemplate id="Artifact_93f8753b-17ab-43c5-8f11-e3ec98fe3224"
      type="sh">
    <ns2:Properties />
    <ns2:ArtifactReferences>
      <ns2:ArtifactReference
       reference="http://.../upload/files/wordpress/install_WordpressPHP.sh" />
    ...
</ns2:ArtifactTemplate>
...
</ns2:Definitions>
```

Figure 19. TOSCA-based Description of WordPress Application

V. The advantages of CAM-D

To evaluate the advantages of CAM-D in particular and CAM [2][3] in general, we make a comparison between CAM-D and TOSCA in Table I because TOSCA has been the well-known standard. However, TOSCA specification has not totally developed for multi-cloud application specification.

TABLE I
THE COMPARISON BETWEEN CAM-D AND TOSCA SPECIFICATION

FEATURES	CAM-D	TOSCA Specification
Topology	Nested structure	Flat structure
Cloud software portability	YES	NO
Component-based cloud application description	YES	NO
Synthesized from component specifications	YES	NO
Multi-cloud service matchmaking	YES	NO

The above comparison showed significant differences of CAM-D compared with TOSCA Specification. CAM-D especially supports Cloud software portability, Component-based cloud application description, Synthesized of component specifications, and Multi-cloud service matchmaking. These features have proved the feasibility of our proposal with distinct advantages for cloud application development as follows:

- A cloud application may be constructed as a distributed system of which elements are compute servers running specific software and located at specific cloud providers.
- Developers are free to evolve their cloud software without any technology restriction from cloud providers. They also can develop and sell their small pieces instead of complete software solutions.
- Enhancing the ability to reuse the cloud software component, developers can build up an application by just incorporating existing components of others into their own software solution. This save cost and time in cloud application development.

In addition, an interesting feature of CAM-D that is quite similar to a program written in a programming language. The application template is the main program, component templates are procedures and arguments of a procedure can refer to parameters of others. This is very convenient for developer to independently develop cloud software. He could package the software component as a "***Black Box***" and just express properties to the outer ports of such component. This is the special feature that CAM-D brings.

VI. Conclusion

In this study, we build a description method (CAM-D) that supports to describe multi-component cloud software. To implement this idea, firstly, we define CAM-based description templates for multi-component cloud software modeled in a nested structure. Secondly, we develop flattening algorithm to cover nested description to flattening description. Thirdly, As an illustration, we experimentally transform the description templates of CAM-D into TOSCA-based specification templates and demonstrate this process in a particular example – the Wordpress Application. Finally, the advantages of CAM-D is expressed. To sum up, the key of our work is to create a uniform for multi-cloud applications. The proposed concepts and templates can totally be applied to multi-cloud marketplace.

References

[1] H.-L. Huynh, H.-D. Nguyen, V.-T. Le, and T.-T. Nguyen, "A Composable Application model for Cloud Marketplace," *Journal of Vietnam Science and Technology*, vol. 16, no. 5, pp. 40–45, 2017.

[2] H.-L. Huynh, H.-D. Nguyen, and T.-V. Le, "Matchmaking for Multi-cloud Marketplace Application," *Journal of Information and Communications*, vol. 2019, no. 1, pp. 31–42, 2019.

[3] H.-L. Huynh, V.-D. Tran, H.-D. Nguyen, Z. Hu, T.-V. Le, and Q.-T. Huynh, "Auto-Updating Portable Application Model of Multi-cloud Marketplace through Bidirectional Transformations System," *International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2019)*, pp. 11-24, Malaysia, Sep 2019. DOI:10.3233/FAIA190035. (SCOPUS)

[4] H.-L. Huynh, H.-D. Nguyen, V.-T. Le, and D.-H. Le, "Towards the cloud marketplace for Multi-cloud infrastructures," *in 18th Vietnam National Conference: Selected issues of information technology and communication*, pp. 100-105, November 2015.

[5] K. Sledziewski, B. Bordbar, and R. Anane, "A DSL-Based Approach to Software Development and Deployment on Cloud," *in 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 414–421, April 2010.

[6] E. Brandtzaeg, S. Mosser, and P. Mohagheghi, "Towards CloudML, a Model-based Approach to Provision Resources in the Clouds," *in 8th European Conference on Modelling Foundations and Applications (ECMFA)*, pp. 18–27, 2012.

[7] G. Baryannis, P. Garefalakis, K. Kritikos, K. Magoutis, A. Papaioannou, D. Plexousakis, and C. Zeginis, "Lifecycle management of service-based applications on multi-clouds," *in Proceedings of the 2013 International workshop on Multi-cloud applications and federated clouds (Multicloud'13)*, pp. 13–20, 2013.

[8] G. Sousa, W. Rudametkin, and L. Duchien, "Automated setup of multi-cloud environments for micro services applications," *in 2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 327–334, 2016.

[9] N. Ferry, F. Chauvel, H. Song, A. Rossini, M. Lushpenko, and A. Solberg, "CloudMF: Model-driven management of multi-cloud applications," *ACM Trans. Internet Technology*, vol. 18, Jan. 2018.

[10] "OpenStack HEAT URL." https://docs.openstack.org/heat/latest/. Accessed: 2019-04-26.

[11] "Juju Charms URL." https://jujucharms.com/. Accessed: 2019-04-26.

[12] G. Tricomi, A. Panarello, G. Merlino, F. Longo, D. Bruneo, and A. Puliafito, "Orchestrated multi-cloud application deployment in openstack with TOSCA," *in 2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–6, 2017.

[13] J. Carrasco, J. Cubo, and E. Pimentel, "Towards a flexible deployment of multi-cloud applications based on tosca and camp," *in Advances in Service-Oriented and Cloud Computing (G. Ortiz and C. Tran, eds.), (Cham)*, pp. 278–286, Springer International Publishing, 2015.

[14] K. Alexander, C. Lee, E. Kim, and S. Helal, "Enabling end-to-end orchestration of multi-cloud applications," *IEEE Access*, vol. 5, pp. 18862–18875, 2017.

[15] K. Saatkamp, U. Breitenbücher , O. Kopp, and F. Leymann, "Topology Splitting and Matching for Multi-Cloud Deployments," *in Service-Oriented Computing-ICSOC 2017 Workshops*, pp. 379–383, 2017.

[16] OASIS, "Topology and Orchestration Specification for Cloud Applications Version 1.0," *Organization for the Advancement of Structured Information Standards*, 2013.

Hoang-Long Huynh received B.S (2008) from Nhatrang University and M.S (2012) from Hanoi University of Science and Technology, Vietnam. His research interests in Cloud Computing.
Email: longlove1232002@yahoo.com

Huu-Duc Nguyen received PhD degree (2006) in Computer Science from Japan Advanced Institute of Science and Technology (JAIST), Japan. He is currently the director of the Center for Data and Computation Technologies, Hanoi University of Science and Technology. His main research topics include compiler construction, high performance computing, distributed systems and big data.
Email: ducnh@soict.hust.edu.vn

Trong-Vinh Le received PhD degree (2006) in Computer Science from Japan Advanced Institute of Science and Technology (JAIST), Japan. He is currently Associate Professor and the director of the Center for Information Technology and Communication, VNU University of Science. His main research topics include theory of algorithms, computer networks.

Email: vinhlt@gmail.com

Quyet-Thang Huynh received PhD degree (1995) in Information and Computer Sciences from Varna Technical University, Bulgaria. He is currently Associate Professor and the president of Hanoi University of Science and Technology. His main research topics include Software Quality, Software Testing, Methods in Software Development, Multi-Objective Optimization, Project Management, Big Data Processing and Analytics.

Email: thanghq@soict.hust.edu.vn

Elasticity for MQTT Brokers in IoT Applications

Linh Manh Pham[1,2,3], Tien-Quang Hoang[4], Xuan-Truong Nguyen[4]

[1] Graduate University of Science and Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet, Cau Giay, Ha Noi, Vietnam

[2] Institute of Information Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet, Cau Giay, Ha Noi, Vietnam

[3] VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

[4] Hanoi Pedagogical University 2, 32 Nguyen Van Linh, Xuan Hoa, Phuc Yen, Vinh Phuc, Vietnam

Correspondence: Linh Manh Pham, linhmp@vnu.edu.vn

Communication: received 7 November 2020, revised 7 December 2020, accepted 31 January 2021

Digital Object Identifier: 10.32913/mic-ict-research.v2020.n2.941

***Abstract*: Many domains of human life are more and more impacted by applications of the Internet of Things (IoT). The embedded devices produce masses of data day after day requiring a strong network infrastructure. The inclusion of messaging protocols like MQTT is important to ensure as few errors as possible in sending millions of IoT messages. This protocol is a great component of the IoT universe due to its lightweight design and low power consumption. Distributed MQTT systems are typically needed in actual application environments because centralized MQTT methods cannot accommodate a massive volume of data. Although being scalable decentralized MQTT systems, they are not suited to traffic workload variability. IoT service providers may incur expense because the computing resources are overestimated. This points to the need for a new approach to adapt workload fluctuation. Through proposing a modular MQTT framework, this article provides such an elasticity approach. In order to guarantee elasticity of MQTT server cluster while maintaining intact IoT implementation, the MQTT framework used off-the-shelf components. The elasticity feature of our framework is verified by various experiments.**

Keywords: *Elasticity, MQTT broker, Cloud computing, Internet of Things.*

I. INTRODUCTION

The Internet of Things (IoT) has had a significant influence on many aspects of life at various scales, ranging from households to enterprises to large organizations or industries. Millions of connected computers, with or without human interference, are behind IoT applications, attempting to communicate and transfer data over the Internet. It is predicted that, by 2050, there will be about 170 billion objects or things connected to the Internet [1]. Additionally, the IoT network can hold between 50 and 100 trillion connected objects, and it can track the movement of each object. Every person living in metropolitan areas can be surrounded and monitored by 1000 to 5000 IoT objects. Currently, by 2020, our world has 4 billion connected people, over 25 million applications, over 25 billion embedded and intelligent systems, and 50 trillion Gigabytes of data generated. This market produces 4 trillion dollars in revenue for service providers [2].

In order to support the communication of billions of IoT devices and to rotate the produced masses of data, IoT service providers need to deploy and maintain robust and scalable network infrastructures. When IoT applications scale beyond the scale of household applications to larger systems such as cities or nations, the number of IoT devices can grow very quickly with unpredictable degrees. That is why IoT infrastructures are being developed that not only have strong fault tolerance but also need to be scalable. Most modern IoT infrastructures today deploy an essential component known as the Message Queuing Telemetry Transport (MQTT) server. These brokers use the MQTT protocol, a Machine to Machine (M2M) open protocol that has been around since 1999. It is an open industry standard issued by OASIS and ISO (ISO/IEC 20922) [3]. Thanks to advantages such as compact size, bandwidth savings, low battery power consumption, and space/time separation, MQTT brokers are increasingly undeniably dominating the field of IoT.

Although MQTT broker solutions recently applied both centralized and distributed approaches to manage millions of incoming connection from end devices in a short amount

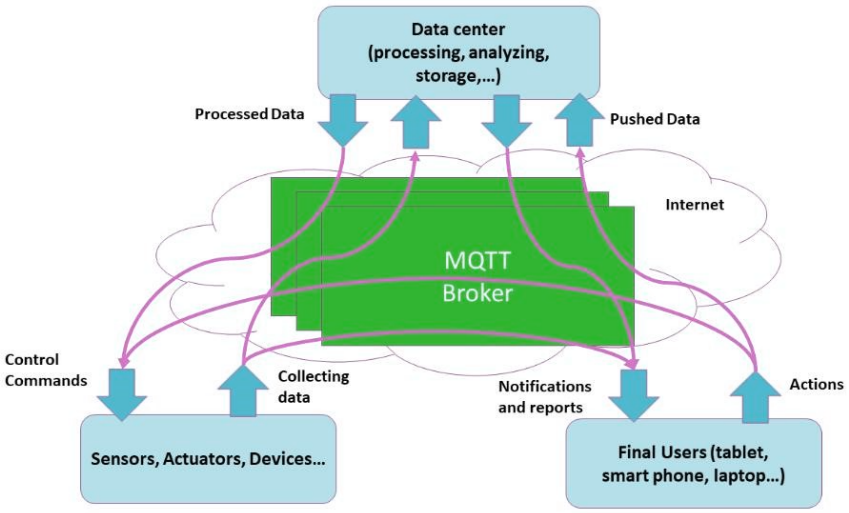

Figure 1. Typical structure of an IoT application using MQTT brokers

of time, there are only several solutions which have functions of adapting to fluctuations in workload generated by loT devices. This variation in load can occur when the number of loT devices fluctuates unpredictably during certain times. In reality, citywide loT applications often deal with dispersed and sporadic terminal equipment, resulting in a shift in the number of devices involved in the application. For example, smart cars are more likely to link to vehicle networks connected during peak hours than during off-peak hours, resulting in the generation of more IoT data during specific times. This necessitates the development of new MQTT-based systems that not only understand how to scale to meet demand but also keep up with changes in workload created by the terminal. In other words, it is a strategy for making MQTT's brokers more flexible.

Elasticity is a fundamental property of Cloud Computing as defined by NIST [4]. Because of scalability, cloud resources are not overused or under-utilized. This does not only save money for the Cloud service provider, but also positively enhances customer experience with the service provided. It is a fact that many IoT applications have already been deployed or are in the process of being deployed on the Cloud. It is worth noting that loT resources such as MQTT brokers deployed on the Cloud may also benefit from this elasticity characteristic of the Cloud.

In this article, we suggest an architectural framework for making MQTT brokers elastic. To achieve this aim, here is what to do:

- We propose a new architecture framework that can flexibly enable elasticity to distributed MQTT brokers while maintaining all the original features of the MQTT protocol.
- We specifically implement the proposed architectural framework using the open-source MQTT brokerage service (EMQX) and private cloud platform (OpenStack).
- We perform experiments to validate the proposed architecture framework using the version deployed with the aforementioned open-source applications.

The rest of the paper is laid out as follows. Section 3 describes in detail the overall architecture of the proposed MQTT architectural framework after highlighting related studies in Section 2. Section 4 discusses implementing the architecture framework using suitable open-source tools. Section 5 presents the validation as well as the results. Finally, in Section 6, we make our conclusions.

II. RELATED WORK

1. MQTT

MQTT is a message-oriented middleware (MOM) software following the publish/subscribe model [5]. To ensure secure transmission in environments with many limitations, IoT applications clearly need an efficient and powerful solution such as the MQTT broker model. Some of the most commonly used MQTT broker solutions today are EMQX, HiveMQ, VerneMQ, Moquette, Mosquitto, etc.

In recent decades, many IoT applications have used MQTT brokers such as [6], [7], [8], [9], [10], [11]. The typical structure of an IoT application using MQTT brokers deployed with remote centralized data centers (as in a Cloud environment) is depicted in Figure 1. The application aims to gather data from a variety of loT devices and sensors, then process and store these data, and finally sends notifications and reports to end-users (using laptops, mobile

devices, tablets, etc.). In certain situations, collected data may be released directly to the topics subscribed to by the end-user without any data analysis. Control commands, like every other form of MQTT message, can be published to the command topic in the brokers. These messages will be stored in cloud archives and transmitted to IoT devices or sensors using some programming mechanisms. In the case of time-sensitive applications, command messages can also be sent directly to IoT devices without having to be sent to the Cloud. We discovered that the IoT devices, end-user interfaces, and data analysis systems are all different types of MQTT clients that produce and ingest remotely measured data.

2. Distributed MQTT broker

Many IoT applications typically implement a centralized MQTT broker capable of keeping all subscribed topics. However, in this model, it is easy for the server to become an obstacle to the entire system. To prevent this, a variety of distributed solutions have been suggested, which can be divided into two categories: a bridged broker and a clustered broker. The two brokers could be bridged in the first model to serve more messages from the client while remaining separate from each other's locations. Published messages are forwarded from a broker to a broker bridged with its specific access policies. A complete mesh network needs to be formed between the brokers (each one must have connections to all the other machines) so that any MQTT client can connect to any brokers that they want. Therefore, using the bridging model to gain the elastic function is too complex. It is only suitable for simple networks which have a few MQTT brokers. The MQTT brokers that support bridging include EMQX, HiveMQ, VerneMQ, Moquette, Mosquitto, JoramMQ, etc. [12]. In the study of Collina et al. [13], Schmitt et al. [14], and Zambrano et al. [15], some implementations of this model have been reported.

MQTT brokers in a cluster model utilize subtopics in a hierarchical structure. One of the brokers (B_0) keeps the original topics and the child topics that have registrations related to them. The other servers (B_1, B_2, etc.) only keep relevant subtopics that originated from B_0. Topic branches operate in a server that corresponds to MQTT registrations for these brokers. As a result, the costs between servers based on the cluster model are significantly reduced compared to the bridging model. Thanks to the brokers' transfer knowledge about the topic and the routing table, any MQTT client may establish or continue their session by connecting or reconnecting to any broker. Only a few MQTT brokers, such as RabbitMQ, VerneMQ, EMQX and HiveMQ [16], currently support the full capabilities of the cluster model. The work of Jutadhamakorn et al. [17], Thean et al. [18], and Detti et al. [19] are some of the studies that follow this trend.

3. Elastic MQTT broker

One of the fundamental properties of Cloud Computing is elasticity. Cloud resources can be provisioned or released when demanded thanks to this unique feature. Today, loT applications are often deployed on the Cloud to utilize the advantage of this environment such as on-demand measurement services, wide-area network capability as well as radically supported resource elasticity. Many solutions, including DOCKERANALYZER [20], Proliot [21], BDAaaS [22] and ACD [23] have attempted to provide scalability for components of IoT applications. However, only a few elasticity solutions for MQTT brokers have been proposed, including Broker [24], E-SilboPS [25]. Many MQTT-specific customizations have been simplified or ignored because Brokel only defines a multilevel elasticity model for brokers through the publish/subscribe model (including MQTT) in general. E-SilboPS is a scalable content-based subscription/publishing system specifically designed to support sensing and contextual communication in IoT-based services. Therefore, it also ignores many of the customized Quality of Service (QoS) parameters of the MQTT protocol and only provides content-based elasticity. One of the prerequisites for a comprehensive elastic MQTT is that the solution must use one of the distributed models mentioned in sub-section II.2.

III. ARCHITECTURE OF ELASTIC MQTT FRAMEWORK

This section presents the proposed elastic MQTT architectural framework. The system is designed with a flexible architecture containing a representative set of modules. When a version of the framework is deployed on the Cloud, each representation module is specialized into a specific open-source software component that is already available. Framework's modules can therefore be flexibly replaced to gain new features, for higher efficiency, or lower software licensing costs. The architecture framework also supports the MQTT clustered broker model to reduce communication costs among the cluster servers. Figure 2 shows the framework's overall architecture, which includes the following modules:

- **MQTT Broker Cluster**: A group of MQTT brokers implements a distributed publish/subscribe model with the customized QoS parameters. The cluster contains several systems that provide an execution environment called a node. The nodes connect to each other using

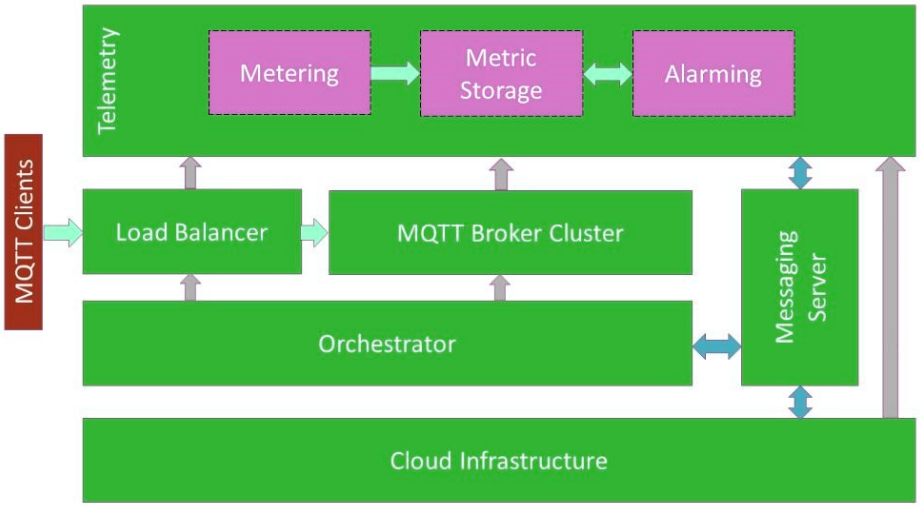

Figure 2. The architecture of Elastic MQTT Framework

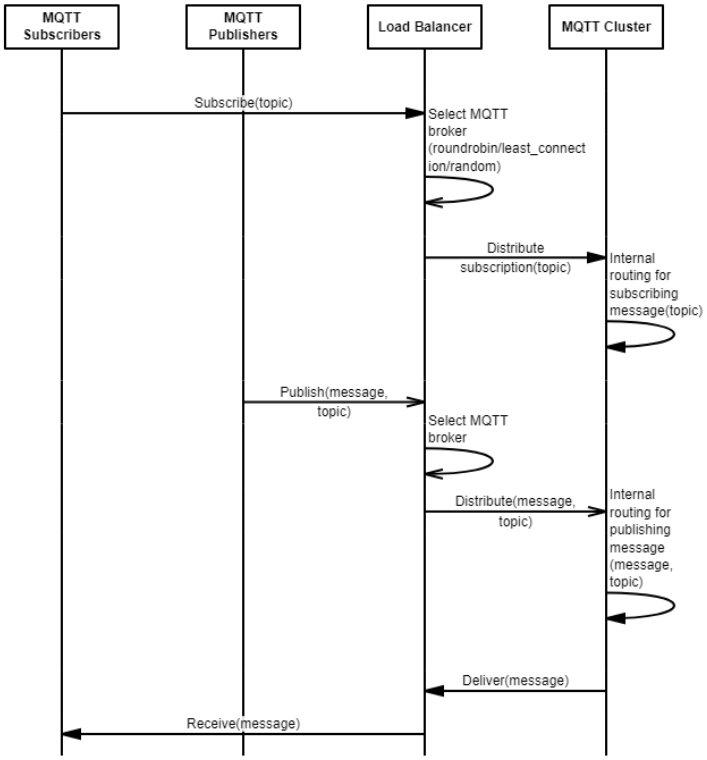

Figure 3. Schematic diagram of data flow during publishing - subscribing messages using load balancer

TCP/IP and communicate by transmitting the message. Each node holds sections related to itself in the topic system and the current subscriptions. This mechanism enables the published messages to be routed over the entire cluster from the first node that receives the message to the last node that sends it to the subscriber. Nodes can join the cluster manually or automatically. With automatic manner, automatic mechanisms for cluster identification and joining such as IP multicast, dynamic DNS or ETCD [26] need to be supported. Nodes can be deployed on either public or private clouds. Public Cloud providers, such as AWS, Azure, or private cloud platforms, such as OpenStack, CloudStack, could all be viable options.

- **Load Balancer**: A Load Balancer (LB) is usually deployed in front of an MQTT cluster to distribute the

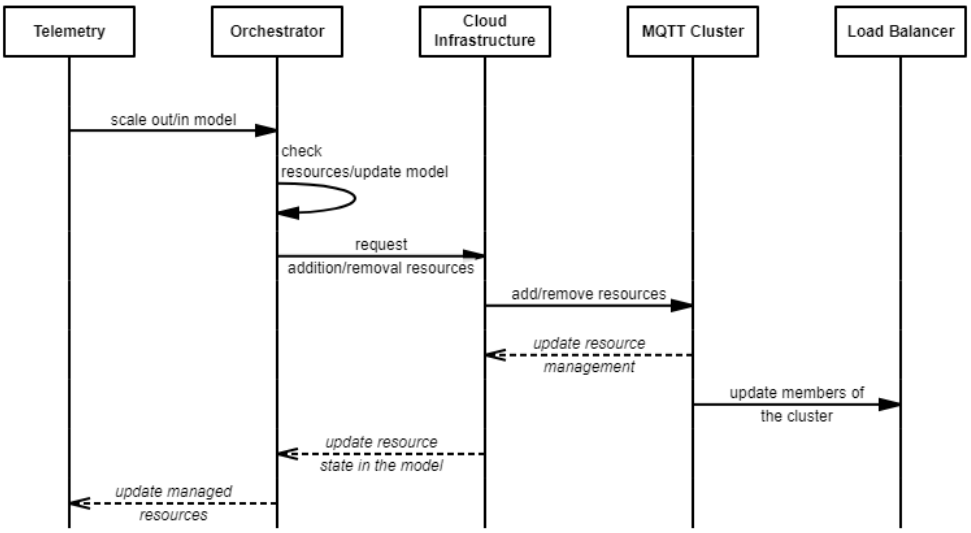

Figure 4. Data flow of components when it has a resource scaling event on MQTT brokers

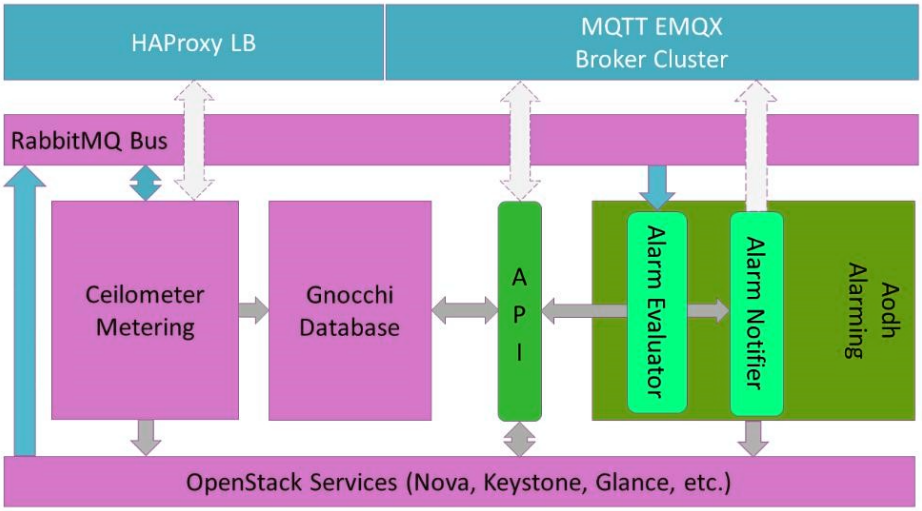

Figure 5. An implementation of the elastic MQTT architectural framework

connections and messages from the devices into the MQTT cluster. LB also enhances the availability of clusters, balances load between nodes in the cluster, and provides dynamic scalability. The links between LB and the nodes in the cluster are simple TCP connections. By doing this, a single MQTT cluster can serve millions of clients. Thanks to LB, the MQTT client only needs to know one connection point instead of maintaining a long list of MQTT brokers. A schematic diagram of the data flow during publishing - subscribing messages using a load balancer is illustrated in Figure 3.

- **Orchestrator**: This module parses descriptions of system components in their domain-specific high-level language (i.e., Domain-Specific Language - DSL) and then deploys, manages, and monitors the full life cycle of all components involved. Those components include resources such as virtual machines, containers, images, security groups, alerts, extension policies, etc. DSL grammar can be derived from XML, JSON, or YAML with the primary motivation to keep things simple and user-friendly. Within the architecture framework, the Orchestrator deploys and manages the MQTT brokers as well as resources of elastic decision-making block such as Metering, Metric Storage, and Alarming.
- **Telemetry**: This module includes three sub-modules: Metering, Metric Storage, and Alarming.

  Metering: The goal of this module is to collect, standardize, and transform data generated by coordinated components effectively. These data will be used to gen-

erate multidimensional views and help resolve different telemetry cases. Among them, data of specific metrics (i.e., measurements) are collected and analyzed for the purpose of decision-making on elasticity. Alarming and Metric Storage are two modules that directly exploit these measures.

Alarming: Its goal is to provide the ability to trigger response actions based on predefined rules relied on sample data or events collected by the Metering module. It consists of two main sub-modules: "Alarm evaluator" and "Alarm notifier". The first module evaluates the measurements of a given metric stored in the Metric Storage module to find out whether they are over or below a predetermined threshold or not. The second module activates the notifications and sends them to the Orchestrator to order the corresponding elastic actions such as increasing/ decreasing the number of MQTT brokers.

Metric storage: A database that mainly stores aggregate measures of cluster nodes such as system performance metrics. A measurement is a list of the form *(timestamp, value)* for a given managed resource. Resources can be anything from the temperature of the nodes to the CPU usage of a virtual machine. Besides, the database also stores events, which are lists of things happening in the Cloud infrastructure such as a request to an API received, a virtual machine started, a photo uploaded, or anything else. Stored measurements are retrieved for monitoring, billing, or alerting purposes, in which saved events are useful for testing, performance analysis, debugging, etc.

- **Cloud infrastructure**: It manages, provides, and dynamically releases virtual resources for scalability. For "unlimited" resources, all private, public, or hybrid clouds may be required to deploy.
- **Messaging server**: It is essential to carry out communication between the modules of the architecture framework by exchanging messages. It creates channels to be connected using popular communication protocols like CoAP, AMQP or even MQTT.

All the modules of the framework are detachable. That means the boot order of the components is not entirely crucial. Even so, some modules working independently usually do not make much sense, so there are still prerequisites for the start-up of other modules beforehand. Likewise, the components and resources are described and managed by *the Orchestrator*, which can be instantiated at any point in time. The Orchestrator must be able to resolve dependencies between the components and therefore come up with a deployment plan that contains the correct order of installation. From the system description to the deployment plan, the Orchestrator needs to use a series of resolvers such as Learning Automata based allocators, Constrained Programming-based solvers, and Heuristics and Meta solvers. When an event or combination of events and conditions occurs during execution, the Orchestrator generates the corresponding scaling plan and makes the necessary modifications to transform the existing implementation model to the proposed deployment model describing in the elastic plan. Modifications include actions that comply with ECA (Event-Condition-Action) laws such as horizontal or vertical scaling of a resource when measurements of the resource violate the thresholds beforehand. Figure 4 illustrates a simplified coordination diagram between the components mentioned in the MQTT architectural framework during the resource elastic event.

IV. THE IMPLEMENTATION

In this section, we present a specific deployment of the architectural framework proposed in the previous section. It mainly supports cloud-based IoT applications that require scaling as an essential feature. These applications include, but are not limited to, applications for big data analysis or applications that are sensitive to latency. With the principle of developing software for the open science community [27], we prioritize the combination of open source solutions that are most suitable for our architectural framework where they are being used universally. Figure 5 depicts a deployment of the architectural framework whose details are described as following.

- **MQTT broker cluster using EMQX**: EMQX is based on the distributed Erlang / OTP programming platform and the Mnesia database [28]. It provides broker nodes that run in parallel and are highly fault-tolerant and distributed. It is one of the few open-source MQTT solutions that support the clustered model. Furthermore, EMQX is the only solution that supports all three levels of MQTT QoS, as well as both MQTT protocols for conventional networks and MQTT-SN for sensor networks. EMQX supports node autodetection and cluster auto-joining with various strategies such as IP multicast, ETCD, dynamic DNS and K8s [29]. In that way, when a broker node comes in or out corresponding to the scaling action, the cluster will automatically recognize the changes and update its configuration to reflect the number of new nodes.
- **Load Balancer**: Several commercial LB solutions that are supported by EMQX such as AWS, Aliyun, or QingCloud. On open-source software, HAProxy [30] can act as an LB for the EMQX cluster and allocate TCP connections. Many dynamic scheduling

algorithms can be assigned on HAProxy such as round robin, least connections, or random.

- **Cloud infrastructure**: To serve the open science community, OpenStack [31], an open-source private Cloud, is chosen to provide and release virtual resources. With a supported worldwide community of users and well-maintained mighty services, OpenStack fits our goals well. Some of the specific OpenStack services used for our deployment are Nova, Keystone, Glance, Horizon, Swift, and Neutron. Because OpenStack cloud is selected, the following modules are officially supported by OpenStack services.
- **Orchestrator**: The official orchestrator supported by OpenStack is Heat service [32]. The infrastructure for a cloud application is depicted in the legibly Heat template file and edited by humans. Infrastructure resources that can be described include servers, storage, users, security groups, floating IP, etc. Heat also provides an automatic elasticity service that integrates with *the Telemetry* sub-modules, so the scaling server group can be included as a resource in the template file. This is a perfect fit for achieving elasticity goals. Sample files can also describe dependencies between resources (for example, this floating IP is assigned to this virtual machine). This helps Heat generate all managed components in the correct order to fully launch the application. Heat manages the entire life cycle of an application and knows how to make the necessary dynamic changes. Finally, it also handles the deletion of all deployed resources when the application completes.
- **Telemetry**: OpenStack has several officially supported services for the Telemetry. These include the Ceilometer service for Metering, Aodh for Alarming, and Gnocchi for Metric Storage.

  Metering: The Ceilometer service [33] provides the following functions: (1) Exploration of measurement data of the OpenStack service, (2) Collecting events and measuring data by tracking notifications sent from the service, (3) Exporting data collected to various defined targets including data warehouse and queue containing record.

  Alarming: When the measurement data or the collected event breaks a given rule, the Aodh service will trigger an alert [34]. The service consists of the following components: (1) An API that provides access to the warning information stored in the metric storage. (2) A alarm evaluator that determines when warnings are triggered due to measurements over a threshold for a period of time. (3) A alarm notifier that allows setting warnings based on an evaluation over a threshold for a collection of patterns.

  The Metric Storage: Gnocchi is an open-source time series database [35]. It attempts to solve storing problems and index time series measurements of OpenStack resources on a large scale. Gnocchi deploys an unique approach to time series storage: instead of storing raw data points, it aggregates (average, minimum, etc.) them before storing them. Since Gnocchi calculates all measurements as aggregate ones when collecting, importing, and processing data, data retrieval is done quickly by re-reading calculated results from before.
- **Messaging server**: In the OpenStack cloud, internal communication between OpenStack services can be performed by RabbitMQ [36]. RabbitMQ is an open source message-oriented middleware that supports popular communication protocols such as STOMP, AMQP and MQTT.

V. EVALUATION

To evaluate and validate the functions of the proposed framework, we performed the implementation described in Section 4 in the data center of VNU University of Engineering and Technology (VNU-UET). We also give some discussions on the results of the experiments.

1. Installing experiment system

The experimental system consists of two main parts: a deployment of our proposed architecture framework and a load generator to simulate multiple MQTT clients and load their work. We create test plans with different scenarios using the available functionality of the load generator. The publishers are facsimiled to create MQTT messages and send them to the LB (HAProxy). In LB turn, it distributes these messages to a cluster of EMQX brokers according to a predefined scheduling algorithm. The simulated subscribers also perform MQTT subscriptions to specific topics in the cluster by connecting to LB. LB also distributes connection requests from subscribers to one of the cluster members. Messages routed from the source to the correct destination are conducted internally in the cluster as mentioned in Section 3. Table I shows the configuration values of the main parameters for HAProxy version 1.8.8 used in the experiment.

We used Apache JMeter [37] version 5.3 as a load generator in our tests. It is an open-source tool to measure the effects of simulation load and evaluate performance. It supports the experiment of various types of protocols like REST, HTTP, HTTPS, JMS, FTP, SOAP, etc. Other protocols can be added to JMeter using plugins. To support the experiments, we developed an MQTT plugin for JMeter

that implements some features of the MQTT 5.0 version [38]. To perform the stress tests, we used a distributed experiment model with a primary JMeter server and a couple of passive JMeter servers. This ensures that there were no side effects on the performance of the MQTT clients simulated.

Our OpenStack private cloud is installed in our data center for research of VNU-UET. EMQX brokers and JMeter load generators are installed in cloud-powered virtual machines. Each EMQX broker instance has 2 vCPU and 2 GB memory. Each JMeter virtual machine has 8 vCPU and 8 GB memory. We use OpenStack Train, which was released in 2019. Our OpenStack Cloud is built on 3 physical servers using an Intel processor. Each physical server has 80 CPU with 2.4 GHz, 256 GB memory and 1.5 TB storage. CentOS 7 is installed on all physical machines as the host operating system. On top of that, KVM is used as a base virtualization solution. For better resource isolation, we install OpenStack controller services (e.g. Nova, Neutron, Keystone, etc.) on the dedicated virtual host instead of running them directly on physical servers. Only the hypervisor service (also known as OpenStack Nova Compute) manages the running of virtual machines, which will be installed directly on the physical servers. In this way, the system services of the OpenStack cloud itself are completely separate from the virtual server instances created by the cloud users. In short, the stress tests performed in our experiments run on OpenStack virtual servers, which will not impact cloud performance and vice versa.

2. Experimental scenarios

We conducted experiments with two common scenarios often found in IoT applications using MQTT: Multi-publisher and Multi- subscriber. Each scenario assesses the effectiveness of the elasticity performance with two models: MQTT centralized broker and MQTT clustered broker.

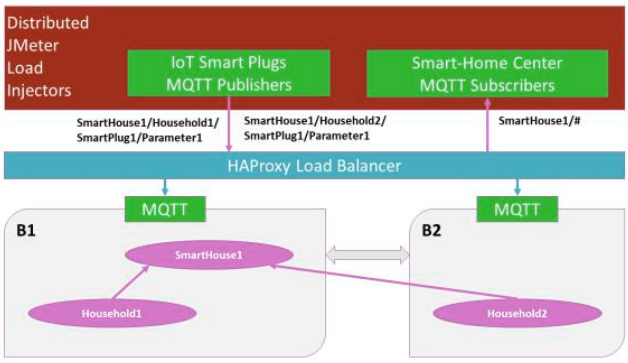

Figure 6. Scenario Multi-publishers on MQTT brokers cluster

TABLE I
MAIN PARAMETERS FOR HAPROXY ARE USED IN THE EXPERIMENTS

Parameters	Value	Meaning
listen	mqtt	Create process "listening" with the responsibility to wait and process MQTT packets.
bind	*:1883	The "listen" process will wait on port 1883 from every network.
mode	tcp	The "listen" process operates in TCP mode.
maxconn	50000	The maximum number of connections per "listen" accepted.
default_backend	emqx_back	The name of the rear broker cluster.
option	clitcpka	Turn on sending TCP keepalive packets on the client-side.
option	srvtcpka	Turn on sending TCP keepalive packets on the server-side.
backend	emq_back	Create a rear broker cluster.
balance	roundrobin	Scheduling mode in turns with round.
timeout check	5000	Set the time to wait for additional check after the connection has been established.
server	emqx_node[i]	Identifies the "i" node that participates in the cluster.

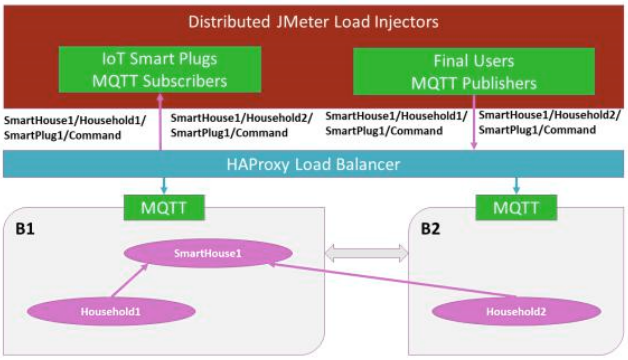

Figure 7. Multi-Subscriber clustered brokers scenario

a) Multi-publisher scenario

This scenario simulates a large number of IoT devices, such as smart plugs, publishing telemetry data to a central smart home system. The plugs are the publishers and the central smart home system is the subscriber. The threads are structured like a tree with three levels. The top-level represents smart houses in a district. The middle level is the households in a smart house. The wireless access point in every household allows smart plugs to connect to the Internet and publish data to a central smart home system.

The last level of the tree is smart plugs that send energy

TABLE II
THE CONFIGURATION PARAMETERS OF THE EMQX BROKER CLUSTER

Parameters	Value	Meaning
cluster.discovery	mcast	Automatically discovers and creates clusters based on UDP multicast.
cluster.mcast.addr	239.192.0.1	EMQX multicast address.
cluster.mcast.ports	4369, 4370	EMQX multicast ports.
cluster.mcast.iface	0.0.0.0	Indicates which discovery service the local IP address should link to.
cluster.mcast.ttl	255	Indicates the Time-To-Live value of multicast.
cluster.mcast.loop	on	Indicates if multicast packets have been sent to local loopback address.
cluster.autoclean	5m	Indicates the time to wait before removing inactive nodes from the cluster.

measurements of devices to Wireless access points. A lower topic level added below is the measure level that represents telemetry parameters such as power consumption (kWh). The experiment defines 40 partitions of topics including:

- 1 root topic for smart house "SmartHouse"
- 3 topics "Household" for each smart house
- 30 topics "SmartPlug" for each household
- 10 topics "Parameter" for each smart plug

Thus, a partition of the topic represents 90 smart plugs and each plug publishes 10 telemetry parameters. We have 3600 smart plugs totals in the whole scenario. The experiment for this scenario is depicted in Figure 6.

b) Multi-subscriber scenario

This scenario also simulates a large number of smart plugs controlled by a central smart home system. These smart plugs are subscribers and the central smart home system is the publisher. "SmartPlug" provides bidirectional communication. The end users and the smart house center can send commands to the plugs. Besides, the smart plugs can also respond to these commands, an indicator of an ON/OFF update, for example. The topics are divided into a couple of levels like Multi-publisher scenario. We also define topic partitions presenting 3600 smart plugs providing a control interface, each partition includes:

- 1 root topic for smart house "SmartHouse"
- 3 topics "Household" for each smart house
- 30 topics "SmartPlug" for each household
- 1 topic "Command" for each smart plug

The experiment for Multi-subscriber is depicted in Figure 7.

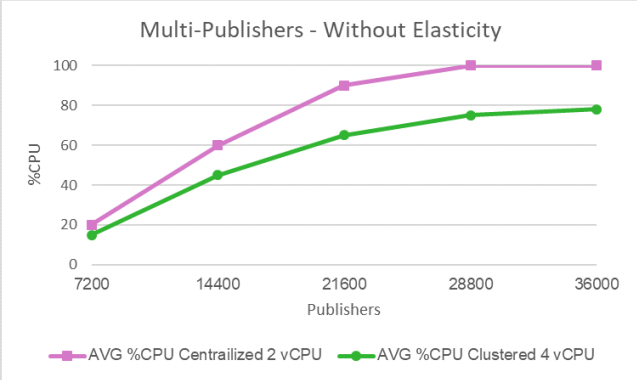

Figure 8. Without Elasticity: Average %CPU Usage of the Multi-Publisher Scenario

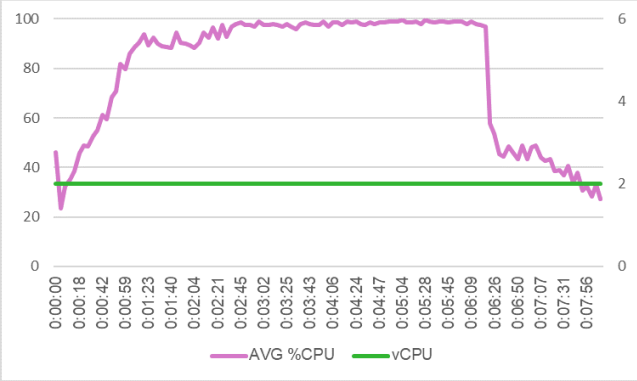

Figure 9. Average %CPU Usage of Multi-Publisher Scenario with the Centralized Broker

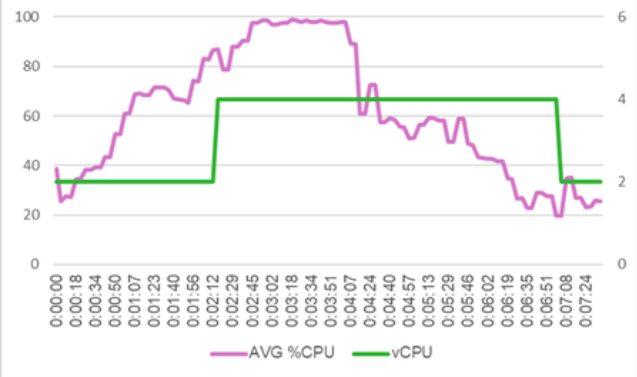

Figure 10. Average %CPU Usage of the Multi-Publisher Scenario with the Clustered Brokers with Elasticity

3. The results

MQTT workloads are created by editing JMeter scripts. The workload starts with a short warm-up and then increases significantly as MQTT clients quickly join the simulation. The EMQX broker is already configured following suggestions from EMQX documentation. We chose IP multicast methods for automatic node-discovery and auto-

join cluster mechanisms. Table II shows the configuration parameters of the EMQX broker cluster. The scheduling algorithm for HAProxy is installed as *roundrobin* (delivered sequentially in turn).

Ceilometer, Aodh, and Gnocchi were configured to measure and store measurements of average %CPU utilization and the number of virtual CPUs (vCPU). Upper and lower thresholds for average %CPU utilization are set to 80% and 25% respectively. It means that if average %CPU utilization breaks the thresholds and the events caught by Ceilometer and Aodh, a notification is sent to Heat for conducting a corresponding elasticity action such as scaling in or out. Actually, Heat has to ask other OpenStack services such as Nova, Keystone or Glance to get the elasticity actions done synchronously. Elasticity plan configured in Heat ensures the number of VMs always in range of 1 to 3.

We used two JMeter client machines for distributed tests. In each client machine, a maximum of 5 JVM processes are allowed to initiate. According to the test scenarios, each process is responsible for running 3600 MQTT clients. Therefore, the maximum 36000 MQTT clients can be started and run in two client machines. To increase saturated probability of the brokers, QoS level of publishing and subscribing MQTT messages is fixed to 2 and "clean session" flag set to FALSE in all experiments.

a) Multi-Publisher scenario

In the case of using a centralized MQTT broker, there is only one subscriber per topic partition. This subscriber listens to all the topics of the partition by subscribing to "SmartHouse/#" with wildcard mask "#" denoting all subtopics of the root topic "SmartHouse". One publisher is created for every topic "SmartPlug" sending messages to the topics "Parameter" below the topic "SmartPlug". In total, 3600 publishers send messages to 36000 topics "Parameter" at a steady rate which is one message/second. In the clustered case, the multi-publisher scenario is tested with a cluster of two brokers B_0 and B_1 (B_1 will be added dynamically when needed). Publishers (90 each partition) and subscribers (one each partition) are equally load-balanced across the two brokers.

Figure 8 shows the average %CPU utilization in both centralized and clustered cases without elasticity. We see that the MQTT system with one broker (2vCPU) is easy to be saturated. Adding one more broker (4vCPU totally) to form the cluster can help resolving the problem. On the other hand, to compare service quality improvement of centralized cases with inelastic clustering using 2 brokers, we performed the experiment 20 times with the average time to complete a scenario which is 6 minutes in the centralized case and 4 minutes 30 seconds in the inelastic clustering case, respectively (down 25%). In both cases, the packet publication rate is halved to avoid brokers saturation which is very likely in centralized cases. In the centralized case, we see in Figure 9 that the average %CPU utilization of the broker reaches saturation after a couple of minutes (at the 1st minute). At this point, the dropped message rate starts to increase. With elasticity, operating cost reduces since we do not have to always maintain multiple brokers (clustered brokers). In Figure 10, we see an elasticity effort to mitigate the pressure performed by our system. One virtual machine of MQTT broker B_1 is created to share the workload. This broker automatically joins the cluster created beforehand by B_0 using the multicast method. The change in the topology is announced to HAProxy for reloading its configuration. The reloading process needs to be used instead of restarting one in order to lower the server downtime as much as possible. After reloading, HAProxy recognizes the new server and distributes messages to all load-balancing members. At the end, the average %CPU utilization of the broker reduces under the lower threshold after a period of time. Thus, we see another elasticity action (scaling in) at this time of the simulation when MQTT clients are finished or terminated. At this point when the workload goes under 25%, the number of VMs is decreased to one for minimizing operating cost.

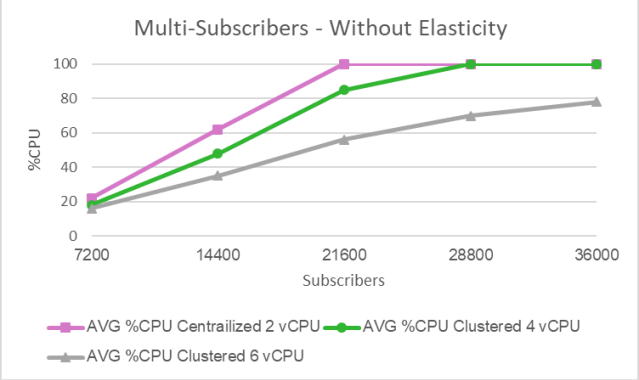

Figure 11. Without Elasticity: Average %CPU Usage of the Multi-Subscriber Scenario

b) Multi-Subscriber scenario

In the case of centralized brokers, there is only one publisher per topic partition. This publisher sends messages to all topics of the partition at a steady speed. A publisher is created for each "SmartPlug" topic. Each publisher receives a message from the "Command" topic under the "SmartPlug" topic. On all the partitions, 3600 publisher received messages from 3600 "Command" topics at a steady speed. In the case of the cluster brokers, the multi-publisher scenario is tested with a group of two MQTT brokers B_0 and B_1 (B_1 dynamically added as needed). The subscribers (90 per partition) and the publisher (each partition has only

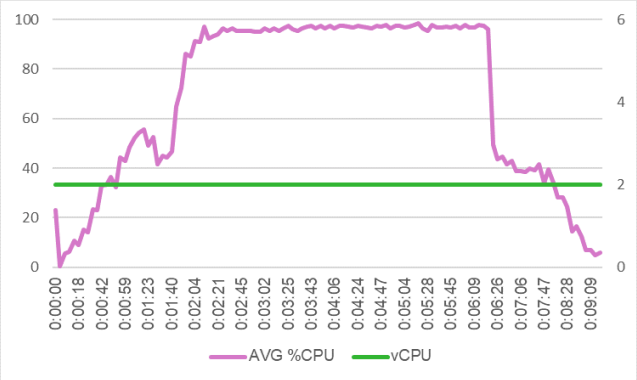

Figure 12. Average %CPU Usage of Multi-Subscriber Scenario with the Centralized Broker

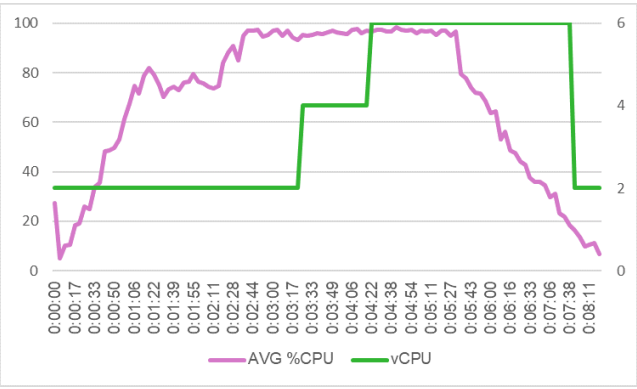

Figure 13. Average %CPU Usage of the Multi-Subscriber Scenario with the Clustered Brokers with Elasticity

one publisher) are load balanced and distributed evenly to the two brokers B_0 and B_1.

Figure 11 shows the average %CPU usage in both centralized and cluster cases which do not have elasticity. We saw the same behavior as that of the multi-publisher scenario. Adding one more broker to the cluster is not really helpful, but two brokers (6 vCPU) might solve the problem. On the other hand, for the comparison of service quality improvement between centralized and inelastic clustering cases using 3 brokers, we performed the experiment 20 times with the average execution time to complete the scenario which is 6 minutes 30 seconds in the case of centralization and 5 minutes 50 seconds in the case of inelastic clustering, respectively (down 10%). In both cases, the message publication rate is halved to avoid brokers saturation which is very easy to occur in centralized case.

In the centralized case, we also see clearly in Figure 12 that the broker's average %CPU usage will saturate after a few minutes (at the second minute). At this point, the proportion of unsolicited discarded messages also begins to increase.

With elasticity, we also see the same behavior shown in Figure 13, which is the same for many publishers. The scaling action with two brokers is triggered later than the multi-publisher scenario. These two brokers are added sequentially by Heat service. A one-minute gap is set between broker additions to avoid rapid fluctuations of resources. Average CPU usage stayed above the threshold for a longer time than the multi-publishers scenario. The reason is that the combination of the QoS 2 level and the "clean session" flag set to FALSE will keep the messages at the brokers for a longer time, so the more the publishers run in parallel, the busier the brokers are.

VI. Conclusion

We have presented a flexible architectural framework that can support scaling functionality for distributed MQTT brokerage services in IoT applications. Our architectural framework provides service elasticity by using open source software that is currently commonly used in the cloud computing environment. Our elasticity MQTT brokerage service has been successfully deployed using EMQX as the MQTT brokers and OpenStack as a private cloud environment. Experiments are conducted by generating the IoT traffic for the service at different load levels to observe changes in the number of broker instances. Our experimental results show that the proposed MQTT brokerage service adapts well to the end-user load changes while maintaining low operating costs.

Acknowledgement

This research is funded by Graduate University of Science and Technology under grant number GUST.STS.ĐT2019-TT02.

References

[1] N. Sharma and D. Panwar, "Green IoT: Advancements and Sustainability with Environment by 2050," *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1127-1132, 2020.

[2] V. Turner, D. Reinsel, J.F. Gantz, S. Minton, "The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things," *Journal of Telecommunications and the Digital Economy IDC Report Apr*, 2014.

[3] MQ Telemetry Transport, "http://mqtt.org," visited on Oct, 2020.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, (2011).

[5] P. Th. Eugster, P. A. Felber, R. Guerraoui, A. Kermarrec, "The many faces of publish/subscribe". *ACM ACM Computing Surveys*, Vol. 35, No. 2, pp. 114-131, 2003

[6] R. Kawaguchi, M. Bandai, "Edge Based MQTT Broker Architecture for Geographical IoT Applications," *2020 International Conference on Information Networking (ICOIN)*, pp. 232-235, 2020.

[7] V. Gupta, S. Khera, N. Turk, "MQTT protocol employing IOT based home safety system with ABE encryption," *Multimed Tools Appl*, vol. 80, pp. 1-19, 2020.

[8] Mukambikeshwari, A. Poojary, "Smart Watering System Using MQTT Protocol in IoT," *Advances in Artificial Intelligence and Data Engineering. Advances in Intelligent Systems and Computing*, vol. 1133, pp. 1415-1424, 2020.

[9] Y. C. See, E. X. Ho, "IoT-Based Fire Safety System Using MQTT Communication Protocol," *ijie*, vol. 12, no. 6, pp. 207-215, 2020.

[10] S. Nazir, M. Kaleem, "Reliable Image Notifications for Smart Home Security with MQTT," *International Conference on Information Science and Communication Technology (ICISCT)*, pp. 1-5, 2019).

[11] P. Alqinsi, I. J. M. Edward, N. Ismail, W. Darmalaksana, "IoT-Based UPS Monitoring System Using MQTT Protocols," *4th International Conference on Wireless and Telematics (ICWT)*, pp. 1-5, 2018.

[12] Comparison of MQTT Brokers, "https://tewarid.github.io/2019/03/21/comparison-of-mqtt-brokers.html," visited on Oct, 2020.

[13] M. Collina, G. E. Corazza, A. Vanelli-Coralli, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST," *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pp. 36-41, 2012.

[14] A. Schmitt, F. Carlier, V. Renault, "Data Exchange with the MQTT Protocol: Dynamic Bridge Approach," *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1-5, 2019.

[15] A. M. Zambrano V, M. Zambrano V, E.L.O. Mejía, X. Calderón H, "SIGPRO: A Real-Time Progressive Notification System Using MQTT Bridges and Topic Hierarchy for Rapid Location of Missing Persons," i*n IEEE Access*, vol. 8, pp. 149190-149198, 2020.

[16] The features that various MQTT servers (brokers) support, "https://github.com/mqtt/mqtt.github.io/wiki/server-support," visited on Oct, 2020.

[17] P. Jutadhamakorn, T. Pillavas, V. Visoottiviseth, R. Takano, J. Haga, D. Kobayashi, "A scalable and low-cost MQTT broker clustering system," *2017 2nd International Conference on Information Technology (INCIT)*, pp. 1-5, 2017.

[18] Z. Y. Thean, V. Voon Yap, P. C. Teh, "Container-based MQTT Broker Cluster for Edge Computing," *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1-6, 2019.

[19] A. Detti, L. Funari, N. Blefari-Melazzi, "Sub-Linear Scalability of MQTT Clusters in Topic-Based Publish-Subscribe Applications," *in IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1954-1968, 2020.

[20] M. H. Fourati, S. Marzouk, K. Drira and M. Jmaiel, "DOCKERANALYZER : Towards Fine Grained Resource Elasticity for Microservices-Based Applications Deployed with Docker," *20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pp. 220-225, 2019.

[21] R.R. Righi, E, Correa, M.M. Gomes, C.A. Costa, "Enhancing performance of IoT applications with load prediction and cloud elasticity," *Future Generation Computer Systems*, Volume 109,P. 689-701, 2019.

[22] L.M. Pham, "Autonomic fine-grained migration and replication of component-based applications across multi-clouds," *in Proc. of 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pp. 5-10, 2015.

[23] M. Nardelli, V. Cardellini, E. Casalicchio, "Multi-Level Elastic Deployment of Containerized Applications in Geo-Distributed Environments," *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 1-8, 2018.

[24] V.F. Rodrigues, I.G. Wendt, R.R. Righi, C.A. Costa, J.L.V. Barbosa, A.M. Alberti, "Brokel: Towards enabling multi-level cloud elasticity on publish/subscribe brokers," *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, 2017.

[25] S. Vavassori, J. Soriano, R. Fernández, "Enabling Large-Scale IoT-Based Services through Elastic Publish/Subscribe". *Sensors*, pp. 17-2148, 2017.

[26] A distributed, reliable key-value store, "https://etcd.io/docs/v3.4.0/," visited on Oct, 2020.

[27] D. Roure, C. Goble, "Software Design for Empowering Scientists," *IEEE Software*, vol. 26, no. 01, pp. 88-95, 2009.

[28] EMQX Broker, "https://docs.emqx.io/broker/latest/en/,' visited on Oct, 2020.

[29] Kubernetes, "https://kubernetes.io/," visited on Oct, 2020.

[30] HAProxy, "https://www.haproxy.com/solutions/load-balancing/," visited on Oct, 2020.

[31] OpenStack: Open Source Cloud Computing Infrastructure, "https://www.openstack.org/," visited on Oct, 2020.

[32] OpenStack Heat, "https://docs.openstack.org/heat/latest/," visited on Oct, 2020.

[33] OpenStack Ceilometer, " https://docs.openstack.org/ceilometer/latest/," visited on Oct, 2020.

[34] OpenStack Aodh, " https://docs.openstack.org/aodh/latest/," visited on Oct, 2020.

[35] Gnocchi - Metric as a Service, "https://gnocchi.xyz/," visited on Oct, 2020.

[36] RabbitMQ, "https://www.rabbitmq.com/," visited on Oct, 2020.

[37] Apache JMeter, " https://jmeter.apache.org/," visited on Oct, 2020.

[38] L.M. Pham, T.T. Nguyen, M.D. Tran, "A Benchmarking Tool for Elastic MQTT Brokers in IoT Applications," *International Journal of Information and Communication Sciences*. Vol. 4, No. 4, pp. 59-67, 2019

Linh Manh Pham is a lecturer at University of Engineering and Technology, Vietnam National University, Hanoi (VNU). He was a postdoctoral researcher at Inria, France. He earned an MSc. in Computer Science in the USA and a Ph.D. in Cloud Computing in France. His area of research is Cloud/Fog Computing, and he intends to highlight the benefits of this relatively novel field of research.
Email: linhmp@vnu.edu.vn.

Tien-Quang Hoang is with Hanoi Pedagogical University 2. He is also a researcher of Center of Multidisciplinary Integrated Technologies for Field Monitoring, University of Engineering and Technology, Vietnam National University, Hanoi (VNU-UET). He has a Master degree in Computer Networks and Data Communication at VNU-UET.
Email:hoangtienquang@hpu2.edu.vn.

Xuan-Truong Nguyen is with Hanoi Pedagogical University 2 as a lecturer. He is also a researcher of Center of Multidisciplinary Integrated Technologies for Field Monitoring, University of Engineering and Technology, Vietnam National University, Hanoi (VNU-UET). He has a Master degree in Software Engineering at VNU-UET.
Email: nguyenxuantruong@hpu2.edu.vn.

Extracting an optimal set of linguistic summaries using genetic algorithm combined with greedy strategy

Lan Pham-Thi[1], Ho Nguyen-Cat[2,3], Phong Pham-Dinh[4]
[1] Faculty of Information Technology, Hanoi National University of Education
[2] Theoretical and Applied Research Institute, Duy Tan University
[3]Faculty of Information Technology, Duy Tan University
[4]Faculty of Information Technology, University of Transport and Communications
Correspondence: Lan Pham-Thi, ptlan@hnue.edu.vn
Communication: received 25 December 2020, revised 26 January 2021, accepted 31 January 2021
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n2.949

***Abstract*: The goal of extracting linguistic data summaries is to produce summary sentences expressed in natural language which represent knowledge hidden in numerical dataset. At the most general level, human users can get a very large number of linguistic summaries. In this paper, we propose a model of genetic algorithm combined with greedy strategy to extract an optimal set of linguistic summaries based on the evaluation measures of goodness and diversity of the set of linguistic summaries. The experimental results on creep dataset have demonstrated the outperformance of the proposed model of genetic algorithm combined with greedy strategy in comparison with the existing genetic algorithm models in extracting linguistic summaries from data.**

Keywords: *Linguistic data summary, hedge algebras, linguistic frame of cognition, genetic algorithm, greedy strategy.*

I. INTRODUCTION

Technology is more and more developed, and data is obtained easily, and their quantity is also increasing rapidly. Therefore, data mining methods are constantly being developed to help people exploit information and knowledge hidden in giant data warehouses. Among those methods, linguistic data summarization is considered a research branch of data mining that has many useful practical applications [1]. The outputs of linguistic data summarization are summary sentences expressed in natural language in a given sentence structure, denoted by *LS* (Linguistic Summary). Each *LS* represents knowledge about the real-world objects stored in a given dataset. The form of knowledge represented in sentences in natural language is easy to understand for every human user. We study the structures of the *LS* used in most studies on linguistic data summarization, which are the sentences with linguistic quantifier proposed by Yager [2]: "*Q y are S*" or "*Q F y are S*" [2–14]. For example, "*Very few* (*Q*) sales of printers (*y*) is with *high* commission (*S*)" [10], "*Most* (*Q*) hospitals (*y*) with *very high* average hospital stay (*F*) have *very low* computer (*S*)" [8].

Human users read the linguistic summaries to understand information and knowledge in the dataset through the semantics of the words '*very few*', '*most*', '*high*', '*very low*', '*very high*' in those linguistic summaries. The linguistic quantifier *Q* represents a proportion satisfying the conclusion *S* with respect to all objects in the dataset in the first sentence sample, or it represents the objects in the group satisfying filter criterion *F* in the second sentence sample.

In the fuzzy set theory approach to extract the linguistic summary from the numerical dataset, the summaries with linguistic quantifier are considered fuzzy propositions expressing knowledge about the objects stored in the dataset. Therefore, the quality of each summary sentence is evaluated at least by a validity measure or a truth value measure. The validity measure formula uses the membership functions of the fuzzy sets representing the semantics of linguistic terms in a sentence like *'very few', 'most', 'high', 'very low', 'very high'* in the above examples. When given a dataset, the results of a linguistic data summarization algorithm are the linguistic summaries with the validity measure greater than a given threshold.

Table I shows a classification of the generality degrees in increasing the order of linguistic data summarization of Kacpryzk and Zadrożny [4]. In which, $S^{structure}$ is the structure of the summarizer *S* as "SALARY = *x*" (*x* is a linguistic term), S^{value} is a linguistic term in the summarizer

S. The degree 5 is the most general degree, when all three components Q, F, S are completely undefined in terms of attributes as well as linguistic terms, then the linguistic data summarization is equivalent to extracting fuzzy rules. Degree 5 poses the big challenge of high computational volume and the large number of linguistic summaries mined from data. However, human users can discover interesting relationships and knowledge hidden in the dataset. The studies in [8, 15–19] applied genetic algorithm to find an optimal set of linguistic summaries. Therefore, human users need to define the constraints and a quality evaluation function for the set of linguistic summaries based on the user's needs. The genetic algorithm will search for an optimal set of linguistic summaries from a set of very large number of linguistic summaries.

TABLE I
CLASSIFICATION OF LINGUISTIC SUMMARY DEGREE

Degree	Given	Search	Note
1	S	Q	Simple summarization by fuzzy query
2	SF	Q	Conditional summarization by fuzzy query
3	$Q\ S^{structure}$	S^{value}	Simple summarization towards determining value
4	$Q\ S^{structure}\ F$	S^{value}	Conditional summarization towards determining value
5	Nothing	$S\ F\ Q$	General fuzzy rules

In the studies on extracting the optimal set of linguistic summaries from the databases by genetic algorithm models [17, 19], in addition to the basic genetic operators (selection, crossover and mutation), the author applied two specific operators. The first one is *Propositions Improver operator* to improve the quality of the linguistic summaries, and the second one is *Cleaning operator* to substitute all propositions in the chromosomes having $T = 0$ with others randomly generated propositions. These two operators were applied into the model of genetic algorithm Hybird-GA to obtain better results than the basic genetic algorithm. However, the experimental results show that there are still limitations of those two operators as follows:

• The *Propositions Improver operator* is applied to substitute an existing linguistic summary with the one selected by a local search based on the best first strategy towards better validity. In the experimental results, there are still 3 out of 30 linguistic summaries with the value of $T < 0.8$, which reduces the evaluation of the goodness of the set of linguistic summaries. This result may be due to the linguistic quantifier set having only five linguistic terms *'none', 'few', 'half', 'much', 'most'*. The fuzzy sets representing the semantics of those five linguistic terms form a strong partition on the universe of discourse, so it is possible to generate a linguistic summary with validity value in the vicinity of 0.5.

• In the experimental results, there is still one out of 30 linguistic summaries with the value of $T = 0$. That is, the *Cleaning operator* has not removed all linguistic summaries having $T = 0$. The linguistic summaries have the value of $T = 0$ because there is not any record in the dataset that satisfies the filter criterion F.

To overcome the aforementioned limitations, we propose a genetic algorithm model that combines the greedy strategy in randomly generating linguistic summaries. The ideas are given as follows:

• We can extend the set of quantifier terms by adding more specificity terms according to the Linguistic Frame of Cognition (LFoC) design method based on the methodology of the hedge algebras as examined in [20]. Since the fuzzy set structure which represents the semantics of the quantifier words [20] is multi-granularity, the higher the specificity level of the quantifier word is set, the higher the chance it is to obtain the linguistic summaries with validity's values closer to 1.

• We only randomly generate the filter criterion F, which is the structure of the summarizer S. We then use the greedy strategy to determine the linguistic term in S and Q such that the validity's value of T and the semantic order of Q are as great as possible. This strategy is applied towards generating favorite linguistic summaries, i.e., increasing the quality measure of the linguistic summaries.

• The linguistic summaries with the value of $T = 0$ are the ones without any record in the dataset that satisfies the filter criterion F. Therefore, in order not show such linguistic summaries, we propose using the support measure $supp(F)$ to evaluate the cardinality of records satisfying filter criterion F. A new linguistic summary is only generated when $supp(F)$ is greater than a given threshold.

To demonstrate the effectiveness of the above mentioned proposals, we perform experiments on the dataset of *creep* and compare the results in the study [19].

The rest of the paper is organized as follows: Section II presents the related issues such as the linguistic summary with linguistic quantifier, the application of genetic algorithm to extract the optimal set of linguistic summaries, and the problem of constructing the fuzzy sets representing semantics of linguistic terms; Section III presents a new proposal of a genetic algorithm model that combines greedy strategy to generate an optimal set of linguistic summaries; Section IV shows the experimental results of one creep dataset and comparative analysis to demonstrate the effectiveness of the new proposals; Some conclusions are presented in section V.

II. SOME FUNDAMENTAL KNOWLEDGE

1. Linguistic summaries with quantifier word

Yager [2] proposes using fuzzy propositions in the structure with linguistic quantifiers expressed in natural language to represent summary information extracted from the dataset. In this section, we briefly present some concepts and notations of the problem of linguistic data summarization.

Let $Y = \{y_1, y_2, \ldots, y_n\}$ be the set of objects (records) in the dataset such as the set of customers of a bank; $A = \{A_1, A_2, \ldots, A_m\}$ is the set of attributes needed to consider objects in the set Y such as AGE, SALARY, MARITAL, etc. We denote $A_i(y_j)$ as attribute value A_i of the object y_j. The dataset is given by the set D = $\{\{A_1(y_1), A_2(y_1), \ldots, A_m(y_1)\}, \ldots, \{A_1(y_n), A_2(y_n), \ldots, A_m(y_n)\}\}$ which is the input of the problem of linguistic data summarization. The output is the linguistic summaries with the linguistic quantifiers having the general structure as follows:

$$Q\ y \text{ are } S \tag{1}$$

$$Q\ F\ y \text{ are } S \tag{2}$$

where:

- *The summarizer* S is an evaluation expressed by a linguistic word in the word domain corresponding to an attribute. For example, AGE = '*young*', SALARY = '*very high*', etc.
- *Linguistic quantifier* Q is a linguistic word representing the proportion of records that satisfies the summarizer S in the entire dataset D like in sentence form (1) or in the object group that satisfies the filter criterion F like the summary sentence of the form (2). For example, '*very few*', '*a half*', '*most*', etc.
- *Validity value* T is a value in the normalized interval [0, 1] evaluating the validity of the linguistic summaries. The value of T is considered the truth value of the fuzzy proposition with linguistic quantifier.
- *Filter criterion* F is optional to define a subgroup of objects in the set of objects Y considered in the linguistic summaries. For example, a fuzzy filter criterion in the form of AGE = '*young*', i.e., only considers the objects in the age group '*young'*.

In a general form, the filter criterion F and the summarizer S are the association of multiple linguistic predicates connected by the connector words AND/OR. Each linguistic predicate is identified by a pair of attribute – linguistic term, such as "AGE = '*young*'", "SALARY =' *high*'", etc. The linguistic summaries in the form of (1), (2) are considered fuzzy propositions with linguistic quantifier. To calculate the truth value of these propositions, it is necessary to design the fuzzy sets representing semantics of the linguistic terms in those propositions. Assume that the semantics of the linguistic terms in the components Q, F, S are represented by the fuzzy sets with respective membership functions μ_Q, μ_F and μ_S. The truth value T is calculated by the formula of Zadeh [21] for the fuzzy proposition with the linguistic quantifier as follows:

$$T(Q\ y \text{ are } S) = \mu_Q\left[\frac{1}{n}\sum_{i=1}^{n}\mu_S\left(y_i\right)\right] \tag{3}$$

$$T(Q\ F y \text{ are } S) = \mu_Q\left[\frac{\sum\limits_{i=1}^{n}\left(\mu_F\left(y_i\right) \wedge \mu_S\left(y_i\right)\right)}{\sum\limits_{i=1}^{n}\mu_F\left(y_i\right)}\right] \tag{4}$$

The truth value T is the basic measure used to evaluate the quality of the linguistic summaries. Therefore, only the linguistic summaries with the truth value of T greater than a given threshold δ are extracted, for example, δ = 0.85 [17] or δ = 0.8 [18]. These are considered linguistic summaries representing information and knowledge hidden in the dataset. In addition, some other evaluation measures are proposed in [4, 22] such as imprecision, covering, focus, and appropriateness. The formulas for calculating these evaluation measures are also based on the membership functions of the fuzzy sets that represent the semantics of linguistic terms appeared in the linguistic summaries.

2. The genetic algorithm extracts the optimal set of linguistic summaries from a given dataset

In several studies of extracting the sets of linguistic summaries from relational databases, we are interested in the research of Donis-Diaz et al. in [17, 19], where the optimal set of linguistic summaries is selected based on *goodness* and *diversity*. The authors coded each linguistic summary into a gene, and each individual is a set of linguistic summaries. In addition to basic genetic operators, the *Propositions improver* operator is added to create an enhanced genetic models [17].

The *Propositions improver* operator uses the neighborhood search technique to replace an existing linguistic summary by a better one. In [19], the *Cleaning operator* is used to replace the linguistic summaries with the truth value of T = 0 by the other randomly generated ones. Two operators - *Cleaning* and *Improver* - are added to the genetic algorithm to make a more efficient hybrid genetic model than the basic genetic algorithms.

Donis-Diaz et al. [17] evaluate the goodness of a good linguistic summary by the value of goodness Gn calculated by formula (5), where T_1, T_2, T_3, T_4 are the truth, imprecision, covering and appropriateness, respectively. Doniz-Diaz et al. in [19] evaluate the goodness of a linguistic summary according to formula (6). In both studies, the goodness of a set of linguistic summaries Gd are calculated by the average of the goodness of the linguistic summaries in the set by formula (7), where l is the number of linguistic summaries in the set.

$$Gn = 0.4 \times T_1^{St} + 0.1 \times T_2 + 0.25 \times T_3 + 0.25 \times T_4 \quad (5)$$

$$Gn = T \times St(Q) \quad (6)$$

$$Gd = \frac{\sum_{i=1}^{l} Gn_i}{l} \quad (7)$$

where $T_1^{St} = T_1 \times St(Q)$ shows the concept of *Linguistic Strength*, and $St(Q)$ is the weight of the linguistic quantifier Q pre-specified based on the priority evaluation of the linguistic quantifiers. Specifically, the parameters used in [17, 19] are St (Most) = 1, St (Much) = 0.75, St (Half) = 0.20, St (Some) = 0.15, St (Few) = 0.05. Thus, the larger proportion the linguistic quantifier represents, the greater the weight.

The diversity of a set of linguistic summaries is calculated by formula (8) in both papers [17, 19], where C is the number of clusters, and l is the number of linguistic summaries in the set.

$$De = \frac{C}{l} \quad (8)$$

C is the number of clusters when clustering linguistic summaries based on the similarity function L as follows:

$$L(p1, p2) = \begin{cases} Yes & if \sum_{k=0}^{m} H(p1_k, p2_k) < 2 \\ No & in\ other\ case \end{cases} \quad (9)$$

The two linguistic summaries $p1$ and $p2$ extracted from the database comprise m attributes represented by a numeric vector consisting of $(m + 1)$ elements. The elements $p1_0$ and $p2_0$ are indexes of the linguistic quantifier Q in $Dom(Q)$, the elements $p1_i$, $p2_i$ are indexes of the linguistic terms in $Dom(A_i)$ of the vector representing the linguistic summaries $p1$, $p2$ ($Dom(A_i)$ - the linguistic domain of the attribute A_i). If the attribute A_i is not present in a linguistic summary, the i^{th} element in the vector representing the linguistic summary takes the value 0. When the result of the function $L(p1, p2)$ is '*yes*', two linguistic summaries $p1$ and $p2$ are similar. The function $H(p1_k, p2_k)$ is calculated by formula (10) to compare the k^{th} element in two vectors. The k^{th} element is different (the value of function $H(p1_k, p2_k) = 1$) when: (1) $p1_k = 0$ and $p2_k \neq 0$; $p1_k \neq 0$ and $p2_k = 0$ (the attribute A_k is present in one linguistic summary, not in the remaining one); (2) the attribute A_k is present in both linguistic summaries, but the index of the two terms are different. Two indices of the terms in the same $Dom(A_k)$ are considered distinct when they are in two positions in ascending semantic order greater than 20% of the number of words in $Dom(A_k)$. For example, if $Dom(A_k)$ = {*'very low', 'low', 'little low', 'medium', 'little high', 'high', 'very high'*}, the term '*low*' at position 2 and the term '*medium*' at position 4 have their distance |2 - 4| > 20%*7 = 1.4. Therefore, two terms '*low*' and '*medium*' are considered distinct.

$$H(p1_k, p2_k) = \begin{cases} 1 & \begin{array}{l} if\ |p1_k - p2_k| > \\ \quad round(0.2 * size(Dom(A_k)) or \\ if\ p1_k = 0\ and\ p2_k \neq 0\ or \\ if\ p1_k \neq 0\ and\ p2_k = 0 \end{array} \\ 0 & in\ other\ case \end{cases} \quad (10)$$

From that, the fitness function Fit of an individual, which corresponds to a set of linguistic summaries, is the weighted sum of two measures: Gd (the goodness of a set of linguistic summaries) and De (diversity) according to the formula as follows:

$$Fit = m_g Gd + m_d De \quad (11)$$

where: m_g, m_d are the weights of two measures Gd and De satisfying the condition $m_g + m_d$ =1. The authors in [19] select m_g = 0.7, m_d = 0.3, i.e., the goodness of the set of linguistic summaries is weighted more than 2 times the diversity of the entire linguistic summaries.

3. Fuzzy set based semantics representation of terms ensures the interpretability of the content of the linguistic summaries

a) The interpretability of the content of the linguistic summaries

In the studies on linguistic data summarization based on fuzzy set theory, the word domain of each attribute is usually limited to 7 ± 2 words, the fuzzy sets representing their semantics are often in the form of strong and uniformly distributed partitions. Figure 1 is an example of the fuzzy set design for the attributes of the patient database in the paper of Almeida et al. [9]. Figure 1(a) includes five fuzzy sets

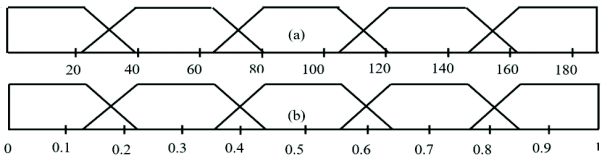

Figure 1. Examples of fuzzy set representation that form strong partitions on the reference domain.

representing the semantics of five terms '*very low*', '*low*', '*medium*', '*high*' and '*very high*' of the attribute "Heard rate". Figure 1(b) includes five fuzzy sets representing the semantics of five quantifier words '*very few*', '*few*', '*half*', '*most*' and '*almost all*'. The linguistic summary extraction algorithm handles directly the membership functions of the fuzzy sets, so they play a decisive role in the output of the algorithm.

Assume that when considering a linguistic summary, denoted by *LS*, extracted from database *D*, the information content assigned to *LS* from the output of the linguistic summary extraction algorithm *M* interacting on the fuzzy sets representing the semantics of linguistic terms in the *LS* is $Cont_{M,D}(fs\_REP(LS))$. The human user interprets the linguistic summary *LS* as a sentence in natural language and receives the information as $Cont_D(LS)$. Determining the conditions to ensure $Cont_{M,D}(fs\_REP(LS)) = Cont_D(LS)$ is considered the problem of examining the interpretability of content of the linguistic summary. However, in the existing studies, the linguistic terms are only considered as linguistic labels assigned to the fuzzy sets designed based on the intuition of the designer of the algorithm *M* and the number of them is limited by 7 ± 2. Thus, when there is no formal formalism to ensure the linkage between the semantics of the linguistic terms and the designed fuzzy sets, it will not ensure that the human user can correctly interpret the content of the linguistic summaries received from the linguistic summary extraction algorithm. To overcome this limitation, in the study [20], the authors have proposed a method of designing fuzzy set structure to ensure the correct semantic representation of terms based on the formalism of hedge algebras theory. In which, the semantics represented by the fuzzy set is designed from qualitative semantics, preserving inherent semantic relationships of the terms in the entire term domain $Dom(A)$ of each attribute *A*. The summary of the method in [20] is as follows:

• Step 1: Determine the syntactic structure, the qualitative semantics of the set of terms $Dom(A)$ of each attribute *A* by a hedge algebras structure. A subset of those terms forms a Linguistic Frame of Cognition (LFoC) $\mathcal{F}_A$ for each attribute *A*.

• Step 2: Based on the inherent semantics of terms in $Dom(A)$, the multi-semantic structure based on the semantic order and the generality - specificity relationships in $Dom(A)$ and $\mathcal{F}_A$ is discovered. Simultaneously, $\mathcal{F}_A$ is scalable in the system development process in practice.

• Step 3: Propose a procedure to design the trapezoidal fuzzy sets from the independent fuzzy parameter set of the hedge algebras structure in Step 1. The trapezoidal fuzzy sets form a structure preserving the multi-semantic structure discovered in Step 2. Simultaneously, the trapezoidal fuzzy set structure of $\mathcal{F}_A$ is also scalable.

b) Multi-semantic structure and the scability of linguistic frame of cognition of each attribute

Based on the viewpoint of hedge algebras, each term domain of the attribute *A*, denoted by $Dom(A)$, is an algebraic structure based on the semantic order relationship (notation ≤) between words. The elements in $Dom(A)$ are induced from two primary terms (considered generator elements) $c^- \leq c^+$, e.g, when considering the attribute *AGE*, c^- = '*young*' and c^+ = '*old*'. The linguistic hedges such as '*very', 'little'*, etc. are used to generate new terms with the semantic order determined based on the semantic change tendency to when the hedges act on the primary terms. We have '*very young*' ≤ '*young*' ≤ '*little young*', but '*little old*' ≤ '*old*' ≤ '*very old*'. Denote the set of hedges by *H*. In general, each term in $Dom(A)$ has the string representation of the form ωc^*, where ω is a sequence of hedges (i.e., $\omega \in H^*$). The length of ωc^* is $|\omega| + 1$, for example, the length of '*very very young*' is 3. In $Dom(A)$, there are 3 constant elements corresponding to the smallest, the neutral and the largest in semantic order. Denote the constant elements are $\boldsymbol{0}, W, \boldsymbol{1}$ respectively; where $\boldsymbol{0} \leq W \leq \boldsymbol{1}$. Since the semantics of the terms need to be determined in the context of the entire $Dom(A)$, the authors in [23] introduce the definition of Linguistic Frame of Cognition (LFoC) of an attribute *A* as follows:

Definition 1: An LFoC of attribute *A*, denoted by $\mathcal{F}_A$, is a set of terms that satisfy the following conditions [23]:

(i) $\{\boldsymbol{0}, c^-, W, c^+, \boldsymbol{1}\} \subseteq \mathcal{F}_A$

(ii) $hx \in \mathcal{F}_A \Rightarrow (\forall h' \in H,\ h'x \in \mathcal{F}_A\ )$

(iii) $x \in \mathcal{F}_A$ và $x = hx'$ $(h \in H) \Rightarrow x' \in \mathcal{F}_A$

From the definition of $\mathcal{F}_A$, each LFoC of an attribute *A* has the form $\mathcal{F}_{A,\kappa} = X_{(\kappa)}$, where $X_{(\kappa)}$ is the set of all words with a length smaller than κ. Then, κ determines the specificity of LFoC. For simplicity, we denote $\mathcal{F}_{A,\kappa}$ as $\mathcal{F}_\kappa$ when *A* is clearly specified.

Based on inherent semantics of the linguistic terms, in $Dom(A) = X$ and $\mathcal{F}_\kappa$ there are the whole semantic order relation ≤ and the generality – specificity partial relation $\mathcal{G}$ between the terms. Two of those semantic relations in conjunction with *X* and $\mathcal{F}_\kappa$ form the multi-semantic

structure $S_{\leq,\mathcal{G}} = (X, \leq, \mathcal{G})$ and $F_{\leq,\mathcal{G}} = (\mathcal{F}_\kappa, \leq, \mathcal{G})$, respectively. Furthermore, when it is necessary to extend LFoC by adding more terms with higher speficity level, we increase the level of κ in LFoC. The semantic order relation and the generality - specificity relation of the existing terms have not been changed. That is, their semantics are not changed when LFoC grows. This scalability is in line with the requirements set out in practice when human users use the set of linguistic terms of attributes in real world perception.

c) Construct the computational semantics to ensure the interpretability and scalability of the linguistic frame of cognition of each attribute

Trapezoidal fuzzy sets are used to represent semantics of linguistic terms in most studies of linguistic data summarization. In the enlarged hedge algebras [24], the interval quantifying mapping value is f: $Dom(A) \rightarrow P[0,1]$ ($P[0,1]$ – which are all subsets of [0, 1]) which allow us to define an interval semantics core of the terms.

The interval semantics core of term x is used as the small base of the trapezoid representing the semantics of x. In [20], the authors propose a procedure for constructing trapezoidal fuzzy sets of terms in $\mathcal{F}_{A,\kappa}$ from the independent semantic parameter set of an enlarged hedge algebra. The trapezoidal fuzzy sets form a multi-granularity structure as illustrated in Figure 2 for a LFoC $\mathcal{F}_{A,3}$, consisting of 3 levels, the terms are induced from the hedge algebras structure with the hedge set $H = \{L\ (\text{'}little\text{'}), V(\text{'}very\text{'})\}$.

The trapezoid set $\mathrm{T}(\mathcal{F}_\kappa)$ is designed to form a multi-granularity structure that preserves the multi-semantic structure of $F_{\leq,\mathcal{G}} = (\mathcal{F}_\kappa, \leq, \mathcal{G})$ and is scalable as LFoC $\mathcal{F}_\kappa$. Hence, they are considered isomorphic images of $\mathcal{F}_\kappa$. These conclusions have been stated into theorems and proved in [20]. We summarize them as follows:

- *Preserving two semantic structural relations*: two inherent semantic-based relations of terms are the semantic order relation $\leq$ and the generality - specificity $\mathcal{G}$. These relations are preserved when mapping from the set of words in $\mathcal{F}_\kappa$ to the set of trapezoid $\mathrm{T}(\mathcal{F}_\kappa)$. We denote $Tr(x)$ as a trapezoid corresponding to the term x. If $x, y \in \mathcal{F}_\kappa$ and $x \leq y$, then $Tr(x) \leq Tr(y)$ because the small base of $Tr(x)$ is at the left of the small base of $Tr(y)$ and at least one of the two end-points of the large base of $Tr(x)$ is smaller than the corresponding one of $Tr(y)$. If $x = hy$ ($h \in H$) (x has the specificity greater than y), then the large base of $Tr(x)$ lies in the inner large base of $Tr(y)$. This is easily seen in the illustration in Figure 2.

- *Scalability*: When LFoC is extended, it means increasing the specificity κ, that is, adding all words that have the specificity level $\kappa + 1$ to the LFoC that has the current specificity level κ. The practical requirement requires that the newly added words do not change the semantics of words already present in the LFoC. With the multi-granularity structure shown in Figure 2, the design of fuzzy sets of words with the specificity level $\kappa + 1$ will not change the fuzzy sets based semantics of words at the specificity level $\kappa + 1$. This is not possible if the fuzzy sets that are based on semantics of LFoC form strong partitions as shown in Figure 1.

Based on the formalism of the hedge algebras theory, we will use the fuzzy sets designed as the proposed method as in paper [20] for the proposed linguistic summary extraction algorithm in Section III and the experiments in Section IV to ensure the interpretability of the content of the linguistic summaries. This is an advantage of the fuzzy set design to ensure the interpretability of the linguistic summary content in this paper, which is not considered in the studies on extracting the optimal set of linguistic summaries by applying the variant of genetic algorithms.

III. GENETIC ALGORITHM COMBINED WITH GREEDY STRATEGY FOR EXTRACTING THE OPTIMAL SET OF LINGUISTIC SUMMARIES

In this study, we propose a genetic algorithm model combined with the greedy strategy to find an optimal set of linguistic summaries extracted from a relational database. The criteria for selecting the optimal set of linguistic summaries in studies [17, 19] are the combination of the goodness and the diversity of the linguistic summary set. This greedy idea is demonstrated in the Random-Greedy-LS procedure to extract an optimal linguistic summary as described in Subsection III.2 below. This procedure is used in genetic algorithm to model genetic algorithm Greedy-GA to extract an optimal set of linguistic summaries as presented in Subsection III.3.

1. Determine the LFoC and the fuzzy sets based semantics of linguistic terms

We use the method of determining the set of linguistic terms for each attribute and designing the trapezoidal fuzzy sets based semantics described in Subsection II.3 to ensure the interpretability of the information content of the linguistic summaries. Firstly, it is necessary to define the syntactic and qualitative semantics of terms in the LFoC $\mathcal{F}_{A,\kappa}$ of attribute A. Information needed include:

- Two generator terms $c^-$and c^+, three term constants $\mathbf{0}$, W, $\mathbf{1}$.
- The linguistic hedges in H and the relative sign table between hedges.

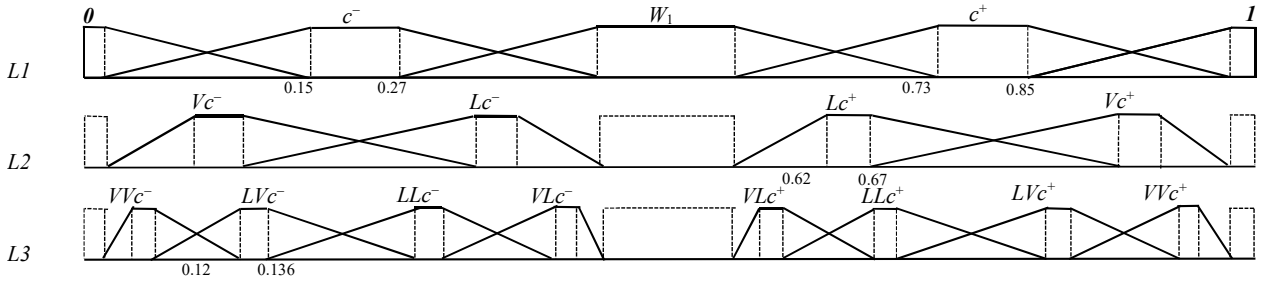

Figure 2. Multi-granularity representing trapezoidal fuzzy sets of terms of an LFoC with 3 levels.

- The speficity level of $\mathcal{F}_{A,\kappa}$, i.e., the value of κ (the maximum length of terms in $\mathcal{F}_A$)

Secondly, to design all trapezoidal fuzzy sets representing the semantics of words in $\mathcal{F}_{A,\kappa}$, it is necessary to provide a set of independent semantic parameter values of the enlarged hedge algebra for attribute A that includes:

- Four fuzzines measure values of three term constants and a generator term c^+ such that they satisfy the constraint: $fm(\mathbf{0}) + fm(c^-) + fm(W) + fm(c^+) + fm(\mathbf{1}) = 1$.
- Fuzziness parameter values of the hedges in H satisfying the constraint $\sum_{h\in H}^{\mu}(h) + \mu(h_0) = 1$, where h is the artificial hedge h_0 which is used to induce the semantics core of the terms.

The set of hedges used in this paper includes a negative hedge '*little*' and a positive hedge '*very*'. In case the specificity level $\kappa = 2$, LFoC has 9 words, i.e., $X_{(2)} = \{\mathbf{0}, Vc^-, c^-, Lc^-, W, Lc^+, c^+, Vc^+, \mathbf{1}\}$. In case the specificity level $\kappa = 3$, LFoC has 17 words, i.e., $X_{(2)} \cup \{VVc^-, LVc^-, LLc^-, VLc^-, VLc^+, LLc^+, LVc^+, VVc^+\}$.

The set of linguistic quantifiers Q plays an important role in extracting linguistic summaries. When the body of the linguistic summary has been specified, that is, the filter criterion F and the summarizer S have been specified, the validity value of the linguistic summary depends on the selection of the linguistic quantifier Q. Since the fuzzy sets are structured in the form of multi-granularity as shown in Figure 2, the set of linguistic quantifier Q consisting of more terms at a greater level of specificity will increase the chance of selecting the linguistic quantifier which allows us to obtain the linguistic summary with higher validity value T.

2. Generate the optimal linguistic summaries based on greedy strategy

Consider the example of an extended linguistic summary in (2): "Q y that (AGE = '*young*' AND SCORE_TEST = '*very high*') are "SALARY = x". In this example, we determine a group of objects satisfying the filter criterion F (AGE = '*young*' AND SCORE_TEST = '*very high*'), the structure of the summarizer S is "SALARY = x", where x is a term in LFoC $\mathcal{F}_{SALARY}$. For each $x \in \mathcal{F}_{SALARY}$, we determine a linguistic summary body, then calculate the value of $r(x)$, which is the proportion of objects satisfying the summarizer "SALARY = x" in the group of objects satisfying the filter condition in the following formula:

$$r(x) = \frac{\sum_{i=1}^{n}(\mu_{young}(o_i) \wedge \mu_{very\_high}(o_i) \wedge \mu_x(o_i))}{\sum_{i=1}^{n}(\mu_{young}(o_i) \wedge \mu_{very\_high}o_i))} \quad (12)$$

where, $o_i \in \mathcal{D} = \{o_1, o_2, \ldots, o_n\}$ is the i^{th} record in the database. From now on, the notation o is used instead of y to avoid confusion with the notation y of the linguistic terms.

Then, choose the term $Q(x)$ such that the validity value $T(x) = \mu_Q(r(x))$ reaches the maximum value. That is, choose the linguistic quantifier that best represents the proportion $r(x)$. When there are many linguistic quantifiers producing the same greatest $T(x)$ value, choose the term with the greatest semantic order.

When $r(x)$ is bigger, $Q(x)$ is chosen with greater semantic order, that is, if $r(x_1) < r(x_2)$ then $Q(x_1) \leq Q(x_2)$. From the formula for evaluating the goodness of a linguistic summary, [19] $Gn = T.St(Q)$, where $St(Q)$ is the priority weight of the terms Q satisfying the condition $Q_1 < Q_2$, then $St(Q_1) < St(Q_2)$. Therefore, if $x^* \in \mathcal{F}_{SALARY}$ then $r(x^*)$ reaches the maximum value, which means there is the largest number of objects in the group satisfying the filter criterion F that SALARY is at x^*, then $Q^* \in Dom(Q)$ is chosen so that the linguistic summary "Q^*y that (AGE = '*young*' AND SCORE_TEST = '*very high*') are SALARY = x^*" has the highest priority $St(Q^*)$ of Q^*.

Referring to the fuzzy association rule, the value of $r(x)$ is the confident measure of the fuzzy association rule of the form "IF (AGE = '*young*' AND SCORE_TEST = '*very*

high') THEN SALARY = x". Therefore, the idea of selecting x^* as in the above example gives a linguistic summary where the summarizer shows the most common property of the attribute SALARY of the object group determined by the filter criterion (AGE = '*young*' AND SCORE_TEST = '*very high*'). When the filter criterion F is completely determined (including the attribute and the corresponding linguistic value), the structure S is determined (the attribute is determined, the term is undefined), we only give out a linguistic summary with the linguistic summarizer x^* that satisfies the following conditions:

(C1): $r(x^*)$ reaches maximum value;

(C2): the validity value T is maximal;

(C3): linguistic quantifier Q^* has maximum order.

Such greedy strategy will make the goodness Gn of each linguistic summary increase, which simultaneously makes the goodness Gn of the whole set of linguistic summaries increase. Moreover, when each group of subjects satisfies the filter criterion F, giving out only one conclusion will also increase the diversity of the set of linguistic summary as evaluated by formula (8). Thus, the quality of the set of linguistic summaries evaluated by formula (11) will also increase.

In the process of extracting the optimal set of linguistic summaries by genetic algorithm, the study [19] used *Cleaning operator* to replace the linguistic summaries with the validity of T = 0 by the other random linguistic summaries. The linguistic summaries with T = 0 correspond to the ones whose filter criterion F includes many AND clauses of linguistic predicates, so there is not any record in the database satisfying the filter criterion F.

The experimental results in [19] show that after an average of 10 runs of the Hybird-GA genetic model using *Cleaning* and *Improver* operators, there is still a linguistic summary with $T = 0$ in the optimal set of linguistic sentences which has 30 sentences in total. In order to not show such linguistic summaries, we propose using the support measure $supp(F) = \sum_{i=1}^{n} \mu_F(y_i)/n$ (n is the number of records in the database) to evaluate the cardinality of the object group satisfying the filter criterion F. We only generate the linguistic summary with the criterion F when the measure support $supp(F)$ is greater than a given threshold β. Thus, the optimal set of linguistic summaries will include summarizers about groups of objects whose cardinality is greater than a threshold β or has the diversity greater than a given threshold.

We use the linguistic summary pattern shown in [20] as follows:

"Qos are $o(E_s)$," and "Qos that are $o(F_q)$ is $o(E_s)$" (13)

where $o(E_s)$ = "$o(A_{s1})$ is/has x_{s1} AND ... AND $o(A_{sm})$ is/has x_{sm}" is the summarizer corresponding to S in (1) and (2); $o(F_q)$ = "$o(A_{q1})$ is/has x_{q1} AND ... AND $o(A_{qh})$ is/has x_{qh}" is the filter criterion in the linguistic summary corresponding to F in (2); A_{ij} is an attribute and x_{ij} is a linguistic term in LFoC $\mathcal{F}_{Aij}$.

From the above analysis, general greedy strategy is applied to select a linguistic summary that is implemented according to the following idea:

- *Step 1*: Randomly generate the filter criterion $o(F_q)$ (include corresponding attribute and linguistic term). Calculate the support measure of $o(F_q)$ by the formula $supp(o(F_q)) = \sum_{i=1}^{n} \mu_{F_q}(y_i)/n$ (n is the number of records in the database). If $supp(o(F_q)) > \beta$, $o(F_q)$ is accepted, go to Step 2. Otherwise, randomly generate another filter criterion $o(F_q)$.

- *Step 2*: Randomly select the attributes in $o(E_s)$ with the given amount, scan the term combinations in LFoC of the attributes $o(E_s)$ to find term combination where the expression $r = \frac{\sum_{o(Fq)}^{\mu} \wedge \mu_{o(Es)}}{\sum_{o(Fq)}^{\mu}}$ reaches the maximum value.

- *Step 3: Select a linguistic quantifier* Q^* in LFoC $\mathcal{F}_Q$ such that the value $T = \mu_{Q^*}(r)$ reaches maximum value. If there are many terms Q^* that make T maximized, select term Q^* with the greatest semantic order.

Step 1 selects the filter criterion $o(F_q)$ satisfying the threshold $supp(o(F_q)) > \beta$, and step 2 selects the term in the summarizer $o(E_s)$ that is the most popular for the object group satisfying $o(F_q)$ according to the defined structure. Step 3 selects the term Q to have the greatest T and the greatest $St(Q)$ (corresponding to the greatest semantic order of Q). It results in the obtained linguistic summary towards larger goodness measure of Gn in the linguistic summaries with the same $o(F_q)$ and the same structure $o(E_s)$.

The procedure for generating linguistic summaries using greedy strategy is described as Algorithm 1.

3. Genetic algorithm combined with greedy strategy for extracting the optimal set of linguistic summaries

a) The object coding in genetic algorithm

Each gene represents a linguistic summary including the following components:

- The filter criterion $o(F_q)$: includes pairs of (ql_i, vq_i), where ql_i is the index of the attribute in the attribute list of the database, and vq_i is the index of the term in LFoC of the attribute at the index ql_i

- The summarizer $o(E_s)$: is similar to the filter criterion F, which includes the pairs of (sm_i, vs_i).

- The linguistic quantifier is the index q_i of linguistic quantifier in LFoC $\mathcal{F}_Q$.

Algorithm 1: Procedure Random-Greedy-LS.

Inputs: Database D, LFoC $\mathcal{F}_A$ và T($\mathcal{F}_A$), where A is an attribute of D, the summary pattern "Qos that are $o(F_q)$ is $o(E_s)$", threshold β.
Outputs: A linguistic summary LS satisfying $supp(o(F_q)) \geq \beta$, the maximum value $r = \frac{\sum^{\mu}_{o(Fq)} \wedge \mu_{o(Es)}}{\sum^{\mu}_{o(Fq)}}$, the maximum value T, Q with the greatest semantic order in the linguistic summaries having the same $o(F_q)$ and $o(E_s)$.
begin
  do
    $o(F_q) \leftarrow$ Random_List((A_{q1}, x_{q1}), …, (A_{qh}, x_{qh}));
  while $supp(F_q) \geq \beta$;
  Random_List(A_{s1}, …, A_{sm});
  $(x_{s1}, \ldots, x_{sm}) \leftarrow (x_{s1}, \ldots, x_{sm}) \in \mathcal{F}_{A_{s1}} \times \ldots \times \mathcal{F}_{A_{sm}}$ AND maximize $r = \frac{\sum^{\mu}_{o(Fq)} \wedge \mu_{o(Es)}}{\sum^{\mu}_{o(Fq)}}$
  $Q \leftarrow Q \in \mathcal{F}_Q$ AND maximize $\mu_Q(r)$ AND maximize $St(Q)$
  return "Qos that are $o(F_q)$ is $o(E_s)$";
end

Algorithm 2: The scheme of genetic algorithm combined with greedy strategy.

for i = 1 to $size\_generation(P)$ **do**
  Add(P, Random-Greedy-LS);
end
evaluate(P);
while *termination criterion not satisfied* **do**
  createEmpty(P');
  add(P', selectElitistChromosomes(P));
  while *P' is not full* ***do*** **do**
    Parent ← select(P);
    Children ← crossover(*Parent*);
    add(P', *Children*);
    evaluate(*Children*);
    end;
  end
  while *mutate criterion satisfied* **do**
    individual ← chooseRandom(P');
    Replace_Genes(*individual*, Random-Greedy-LS);
    end;
  end
  $P \leftarrow P'$;
  end;
end

- The truth value T of linguistic summary.

Each individual (chromosome) represents a set of linguistic summaries consisting of many different genes. Each generation consists of many different individuals.

TABLE II
AN ILLUSTRATION OF THE STRUCTURE OF A GENE REPRESENTS A LINGUISTIC SUMMARY.

q_i	(ql_1, vq_1)	(ql_2, vq_2)	….		(sm_1, vs_1)	(sm_2, vs_2)	…	T

b) The fitness function

The fitness function *Fit* of each individual that represents a set of linguistic summaries is a weighted aggregate measure of two measure: the goodness of the linguistic summary *Gd* in formula (7), and the diversity of the set of linguistic summaries *De* in formula (8). The value of fitness function *Fit* is a value in the interval [0, 1] calculated by formula (11). The best individual in the last generation is selected as the solution to the problem when that individual has the greatest value of *Fit*.

The weight of linguistic quantifier $St(Q)$ in the linguistic summary as in studies [17, 19] is applied to the linguistic quantifiers with the specificity level 1 in the set {**0**, *few, a half, many,* **1**}. The linguistic quantifiers with the specificity level 2 and level 3 are assigned the weighted value such that: if two linguistic quantifiers Q and Q' satisfy the condition $Q < Q'$, then $St(Q) < St(Q')$.

c) The genetic operators

The basic genetic operators are used as follows:

- Selection operator: with each evolution, a proportion of the best individuals (the fitness value *Fit* is greatest) in the current generation are selected for the next generation.

- The crossover operator: one point crossover operator is applied to two randomly selected individuals to generate two offspring. The crossover operator swaps genes between two individuals, i.e. swaps two linguistic summaries between two sets of linguistic summaries. Crossover operator changes the diversity measure of the sets of linguistic summaries, not the goodness of each linguistic summary, but the goodness of the set of linguistic summaries.

- The mutation operator: a small proportion (usually around 0.05) alters some of the genes in a randomly selected individual with a newly randomly generated gene, i.e., replacing some linguistic summaries in the set of summaries. The mutation operator changes both the goodness *Gd* and the diversity *De* of the set of linguistic summaries.

The genetic algorithm scheme combined greedy Greedy-GA strategy is shown in Algorithm 2. In which, the procedure Random-Greedy-LS (as shown in Algorithm 1) is used to generate a linguistic summary using the greedy strategy as presented in Section III.2.

In the genetic algorithm model Greedy-GA proposed in this paper, at the initial generation step, all linguistic summaries (the genes of individuals) are generated by the procedure using the greedy strategy Random-Greedy-LS. In the process of applying the genetic operators to the evolutionary cycles, the selection and crossover operators do not change the linguistic summaries. They just swap the linguistic summaries between the different sets of linguistic

summaries. The mutation operator replaces some linguistic summaries with new ones which are also generated by the procedure Random-Greedy-LS. Thus, all linguistic summaries in the entire executing process of algorithm Greedy-GA are generated by the procedure Random-Greedy-LS. As analyzed in Section III.2, these linguistic summaries tend to increase the fitness function values of the individuals. That is, the results of the proposed algorithm Greedy-GA will give a set of better linguistic summaries according to the adaptability evaluated by the fitness function *Fit* as in formula (11).

IV. EXPERIMENT

In the experiment, we implement the proposed genetic algorithm that combines greedy strategy model Greedy-GA as presented in section III. To demonstrate the advantages of the new proposal in Greedy-GA, the database *creep* is used. The summary patterns and the genetic algorithm parameters are the same as in the model hybrid-GA in study [19] to compare and evaluate the results of extracting the optimal set of linguistic summaries.

1. Database and sentence patterns

The database used in the experiment is *creep* of steel annealing as in the study of Donis-Diaz [19]. The database includes 2066 records and 30 attributes. In which, the attribute *CREEP* represents the strength of steel. There are 19 attributes of chemicals in steel and 6 attributes of temperature. The linguistic summaries are extracted in the form of sentences as in [19], as follows:

- The filter criterion F is a combination of pairs of (*att, val*), each pair (*att, val*) representing a linguistic predicate, where *att* is an attribute in 19 attributes of chemicals or 6 attributes of temperature. The authors in [19] have shown that when the filter criterion F has more than 6 pairs (*att, val*), there is almost no record that satisfies the filter criterion F. Therefore, in this experiment, F will be a combination of no more than 6 pairs of (*att, val*).

- The summarizer S is in the form of 'CREEP= x', where x is a term in LFoC $\mathcal{F}_{CREEP}$.

2. Linguistic of cognition of the attributes and the linguistic quantifiers Q

We use a simple hedge algebra structure which includes two generator terms, 3 term constants, a negative hedge '*little*', and a positive hedge '*very*'. The linguistic frame of cognition of the attributes is $\mathcal{F}_{A,3}$, which consists of 3 specificity levels and 17 linguistic terms. Fuzzy sets representing semantics of terms are represented by the multi-granularity structure shown in Figure 2.

The attribute *CREEP* has a value domain of [13, 550], and the studies in [5, 19] have shown that values between 330 and 550 are considered ideal. The authors used 9 trapezoidal fuzzy sets to represent semantics of 9 terms in *Dom*(*CREEP*), in which the fuzzy set representing the term '*ideal*' (the term has the greatest semantic order in *Dom*(*CREEP*)) has a small base in the interval from 330 to 550, the remaining 8 fuzzy sets are distributed uniformly in the interval from 13 to 330. Thus, we select the following set of parameters for the attribute *CREEP* as follows: *fm*(***0***) = 0.0195; *fm*(*low*) = 0.2832; *fm*(*medium*) = 0.0273; *fm*(*high*) = 0.2793; *fm*(***1***) = 0.3906; $\mu(L) = 0.4$; $\mu(h_0) = 0.25$; $\mu(V) = 0.35$. Then, the trapezoid represents semantics of the term ***1*** (the word with the greatest semantic order in LFoC $\mathcal{F}_{CREEP}$) has a small base coinciding with the small base of the trapezoid representing semantics of the term '*ideal*' in [5, 19]. The trapezoids representing the semantics of the term ***0***, *low*, *medium*, *high,* form a uniform partitions in the interval from 13 to 330 of the reference domain.

The trapezoidal fuzzy sets represent the semantics of terms of the attributes of time, temperature, and chemicals in [19] that form the strong partitions on their reference domains. Therefore, we choose a balanced fuzziness parameter value set for those attributes as follows: *fm*(***0***) = 0.03; *fm*(*low*) = 0.42; *fm*(*W*) = 0.1; *fm*(*high*) = 0.42; *fm*(***1***) = 0.03; $\mu(L) = 0.4$; $\mu(h_0) = 0.25$; $\mu(V) = 0.35$.

The set of fuzziness parameter value of the set of linguistic quantifier Q is determined as follows: *fm*(***0***) = 0.03; *fm*(*few*) = 0.42; *fm*(*a half*) = 0.1; *fm*(*many*) = 0.42; *fm*(***1***) = 0.03; $\mu(L) = 0.4$; $\mu(h_0) = 0.25$; $\mu(V) = 0.35$.

In this experiment, we use the linguistic frame of cognition with the specificity of 3. That is, there are 17 terms in LFoC for each attribute in the database creep and the LFoC of the linguistic quantifier Q. The number of terms of 17 is more than twice the number of terms of the attributes in studies [5, 19].

3. The parameters of genetic algorithm

The parameters of genetic algorithms are selected as in study [19]. Specifically, the number of linguistic summaries in each summary set is 30, corresponding to the 30 genes in each individual. The number of individuals in each generation is 20, and the number of iterations is 100. The selection rate is 0.15, and the mutation rate is 0.1. The fitness function *Fit* evaluates each individual in equation (11) with parameter values $m_g = 0.7$, $m_d = 0.3$.

4. Experimental results

Figure 3 shows the change of the best value of the fitness function *Fit* of the best individual in each generation

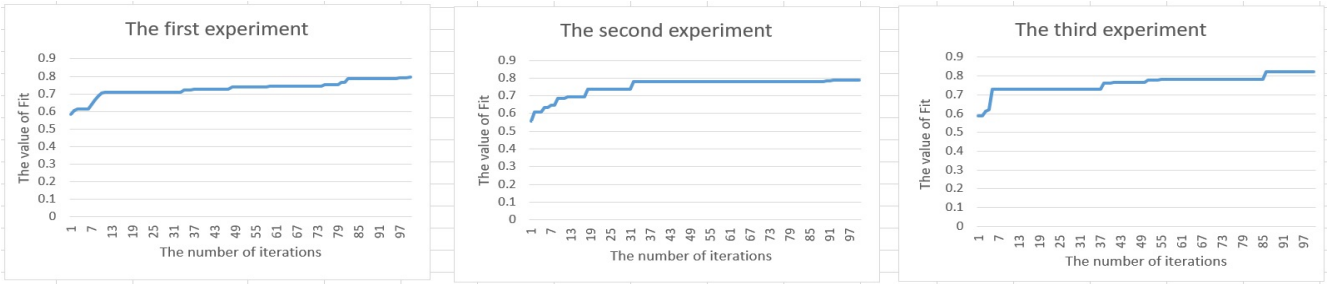

Figure 3. Fitness function value Fit of the best individual in the population in over 100 iterations.

TABLE III
COMPARISON OF THE PERFORMANCE RESULTS OF THE GREEDY-GA MODEL IN THIS STUDY WITH THE HYBRID-GA MODEL IN [19].

Model GA	**Fitness function value *Fit***	**The average of truth values *T***	**The number of summaries with *Q* > *a half***	**The number of summaries with *T* > *0.8***	**The number of summaries with *T* = 0**
Hybrid-GA .[19]	0.6659	0.9139	17.8	27.0	1.0
Greedy-GA	0.7905	0.9951	18.8	30	0

over each loop. From there, it shows that this value has increased gradually and will converge to a value at the last iterations. It proves that the results reflect the evolution through iterations.

Table III shows the comparison between the results of the algorithm Greedy-GA proposed in this paper and those of the algorithm Hybird-GA in the paper of Donis-Diaz et al. [19] on the evaluation function value *Fit*, the average of the validity values T of the linguistic summaries, the number of linguistic summaries with linguistic quantifier with semantic order greater than '*a half*', the number of linguistic summaries with the truth value $T > 0.8$, and the number of linguistic summaries with truth value $T = 0$ (corresponding to the case that there is not any records satisfying condition in F). The model Hybrid GA has been evaluated to be better than the basic model GA (Classical-GA) and the basic model GA combined with the *Cleaning operator* (Classcial + Cleaning-GA) to remove summaries with the truth value $T = 0$. Table III shows that the model Greedy-GA in this study has some advantages in comparison with the model Hybrid-GA:

• The optimal set of linguistic summaries has the greater fitness function value *Fit*. It proves that Greedy-GA gives a better optimal solution.

• There are more summaries which have linguistic quantifiers with semantic order greater than '*a half*'. This is the result of using the greedy strategy for selecting linguistic quantifiers with semantic order as large as possible in linguistic summaries with the same filter criterion F.

• The number of linguistic summaries with truth value T > 0.8 in our experiment reaches the maximum of 30, higher than the result of 27 summaries in [19]. This is because we use the linguistic quantifier term set with 17 terms, and the trapezoids representing the semantics of the linguistic quantifiers form the fuzzy partitions in the form of multi-granularity structure. This proves the advantage of the trapezoidal semantic representation designed based on the hedge algebras theory in [20] and the meaning of the scalability of LFoC in practical applications. Specifically, increasing the number of linguistic quantifiers by using more terms with a larger level of specificity will increase the ability to represent by quantifier term for any proportion in the interval of [0, 1]. The experimental results show that when LFoC of Q consists of 3 levels, it has the ability to select quantifier term for linguistic summaries with truth values greater than 0.8.

• There is no linguistic summary with the truth value $T = 0$. As analyzed at the end of Section III, all linguistic summaries in the execution process of genetic algorithm are generated by the procedure Random-Greedy-LS. Because we have used the condition $supp(F) > 0.1$ for the support measure in the procedure Random-Greedy-LS, there will be no summary with $T = 0$ in the execution of the algorithm Greedy-GA.

V. CONCLUSIONS

The extraction of knowledge represented in words in natural language from numerical datasets is a current trend. In this paper, we proposed a genetic algorithm model combined with the greedy strategy Greedy-GA to extract the optimal set of linguistic summaries based on the evaluation of the goodness and diversity of the set of linguistic summaries. The use of greedy strategy led to the generation of linguistic summaries with high goodness level and the increase of diversity in the set of linguistic summaries. Therefore, the Greedy-GA has better efficiency

when applied to extracting the optimal set of linguistic summaries compared to some existing genetic models. One difference from the existing genetic models to extract the optimal set of linguistic summaries is that we utilize hedge algebras methodology for designing the fuzzy sets based on semantics of linguistic terms. This ensures the interpretability of the content of the linguistic summaries, and that the set of terms consists of many high specificity terms also increase the quality of the set of linguistic summaries. The experimental results of dataset *creep* have proved the effectiveness of fuzzy set design method based on hedge algebras methodology and the model of genetic algorithm combined with greedy strategy in comparison with their counterparts.

REFERENCES

[1] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: a survey", *IEEE transactions on neural networks*, vol. 13, no. 1, pp. 3-14, 2002.

[2] R. R. Yager, "A new approach to the summarization of data", *Information Sciences*, vol. 28, no. 1, pp. 69-86, 1982.

[3] J. Kacprzyk, R. R. Yager, and S. Zadrożny, "A fuzzy logic based approach to linguistic summaries of databases", *International Journal of Applied Mathematics and Computer Science*, vol. 10, no. 4, pp. 813-834, 2000.

[4] J. Kacprzyk and S. Zadrożny, "Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools", *Information Sciences*, vol. 173, no. 4, pp. 281-304, 2005.

[5] C. A. Donis-Díaz, R. Bello-Pérez, and E. V. Morales, "Using Linguistic Data Summarization in the study of creep data for the design of new steels", *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2011, pp. 160-165: IEEE.

[6] A. Wilbik, J. Keller, and J. C. Bezdek, "Generation of prototypes from sets of linguistic summaries", *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2012, pp. 1-8: IEEE.

[7] J. Kacprzyk and S. Zadrożny, "Linguistic summarization of the contents of Web server logs via the Ordered Weighted Averaging (OWA) operators", *Fuzzy Sets and Systems*, vol. 285, pp. 182-198, 2016.

[8] T. Altintop, R. R. Yager, D. Akay, F. E. Boran, and M. Ünal, "Fuzzy Linguistic Summarization with Genetic Algorithm: An Application with Operational and Financial Healthcare Data", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 04, pp. 599-620, 2017.

[9] R. J. Almeida, M.-J. Lesot, B. Bouchon-Meunier, U. Kaymak, and G. Moyse, "Linguistic summaries of categorical time series for septic shock patient data", *IEEE International Conference on Fuzzy Systems (FUZZ)*, 2013, pp. 1-8: IEEE.

[10] J. Kacprzyk and R. R. Yager, "Linguistic summaries of data using fuzzy logic", *International Journal of General System*, vol. 30, no. 2, pp. 133-154, 2001.

[11] M. D. Peláez-Aguilera, M. Espinilla, M. R. Fernández Olmo, and J. Medina, "Fuzzy linguistic protoforms to summarize heart rate streams of patients with ischemic heart disease", *Complexity*, vol. 2019, 2019.

[12] A. Duraj, P. S. Szczepaniak, and L. Chomatek, "Intelligent Detection of Information Outliers Using Linguistic Summaries with Non-monotonic Quantifiers", *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2020, pp. 787-799: Springer.

[13] A. Jain, M. Popescu, J. Keller, M. Rantz, and B. Markway, "Linguistic summarization of in-home sensor data", *Journal of biomedical informatics*, vol. 96, p. 103240, 2019.

[14] A. Wilbik, I. Vanderfeesten, D. Bergmans, S. Heines, and W. van Mook, "Linguistic summaries for compliance analysis of a glucose management clinical protocol", *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018, pp. 1-7: IEEE.

[15] J. Kacprzyk, A. Wilbik, and S. Zadrożny, "Using a genetic algorithm to derive a linguistic summary of trends in numerical time series", *International Symposium on Evolving Fuzzy Systems*, 2006, pp. 137-142: IEEE.

[16] F. E. Boran and D. Akay, "A generic method for the evaluation of interval type-2 fuzzy linguistic summaries", *IEEE transactions on cybernetics*, vol. 44, no. 9, pp. 1632-1645, 2013.

[17] C. A. Donis-Díaz, R. Bello-Pérez, and J. Kacprzyk, "Linguistic data summarization using an enhanced genetic algorithm", *Czasopismo Techniczne*, vol. 2013, no. Automatyka Zeszyt 2 AC (10) 2013, pp. 3-12, 2014.

[18] R. Castillo Ortega, N. Marín, D. Sánchez, and A. G. Tettamanzi, "Linguistic summarization of time series data using genetic algorithms", *EUSFLAT*, 2011, vol. 1, no. 1, pp. 416-423: Atlantis Press.

[19] C. A. Donis-Díaz, A. G. Muro, R. Bello-Pérez, and E. V. Morales, "A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data", *Expert Systems with Applications*, vol. 41, no. 4, pp. 2035-2042, 2014.

[20] C. H. Nguyen, T. L. Pham, T. N. Nguyen, C. H. Ho, T. A. Nguyen, "The linguistic summarization and the interpretability, scalability of fuzzy representations of multilevel semantic structures of word-domains", *Microprocessors and Microsystems*, vol. 81, Article 103641, 2021.

[21] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages", *Computers & Mathematics with applications*, vol. 9, no. 1, pp. 149-184, 1983.

[22] A. Wilbik, "Linguistic summaries of time series using fuzzy sets and their application for performance analysis of investment funds", *Ph. D. dissertation, Syst. Res. Inst.*, Polish Academy Sci., 2010.

[23] C. H. Nguyen, V. T. Hoang, and V. L. Nguyen, "A discussion on interpretability of linguistic rule based systems and its application to solve regression problems", *Knowledge-Based Systems*, vol. 88, pp. 107-133, 2015.

[24] C. H. Nguyen, T. S. Tran, and D. P. Pham, "Modeling of a semantics core of linguistic terms based on an extension of hedge algebra semantics and its application", *Knowledge-Based Systems*, vol. 67, pp. 244-262, 2014.

Lan Pham-Thi was born in Hanoi, Vietnam, in 1984. She received a B.S. in informatics teaching from Hanoi National University of Education in 2006 and an M.S. in computer science from Hanoi National University of Education in 2008. She is currently pursuing a PhD in computer science at the Graduate University of Science and Technology, Vietnam Academy of Science and Technology. Since 2008, she has been working as a lecturer in the Faculty of Information Technology, Hanoi National University of Education. Her research interests include data structure and algorithms, applications of hedge algebra, and approaches to extracting linguistic summaries.
Email: ptlan@hnue.edu.vn

Ho Nguyen-Cat was born in Hanoi, Vietnam, in 1941. He received a B.S. degree in mathematics from the University of Hanoi, the former VNU University of Science of Vietnam National University, Hanoi; a Dr. degree in mathematics and mechanics from the Warsaw University, Warsaw, Poland, in 1971; and a Dr. Sc. degree in mathematics, cybernetics and computer science from the Dresden University of Technology, Dresden, Germany, in 1987. He is now a researcher at the Institute of Theoretical and Applied Research, Duy Tan University in Da Nang and Hanoi. He is the author of more than 90 articles, including more than 40 articles published in international journals and conferences. His research interests include fuzzy logic, fuzzy databases, and computing with words, especially hedge algebras as a mathematical basis for directly handling linguistic words and their applications.
Email: ncatho@gmail.com

Phong Pham-Dinh received a Master degree in Information Technology and a Doctor of Philosophy degree in Computer Science from University of Engineering and Technology, Vietnam National University, Hanoi in 2011 and 2018, respectively. Now, he is a lecturer at the University of Transport and Communications. His research interests include hedge algebras, fuzzy systems, soft computing, data mining, and machine learning.
Email: phongpd@utc.edu.vn

3D Hand Pose Estimation in Point Cloud Using 3D Convolutional Neural Network on Egocentric Datasets

Van-Hung Le[1], Hai-Yen Tran[2]
[1] Tan Trao University, Vietnam
[2] Vietnam Academy of Dance, Vietnam
Correspondence: Van-Hung Le, email: van-hung.le@mica.edu.vn
Communication: received 26 November 2020, revised 27 December 2020, accepted 31 January 2021
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n2.936

***Abstract*: 3D hand pose estimation from egocentric vision is an important study in the construction of assistance systems and modeling of robot hand in robotics. In this paper, we propose a complete method for estimating 3D hand pose from the complex scene data obtained from the egocentric sensor. In which we propose a simple yet highly efficient pre-processing step for hand segmentation. In the estimation process, we used the Hand Point Net (HPN), V2V-PoseNet (V2V), Point-to-Point Regression PointNet (PtoP) for fine-tuning to estimate the 3D hand pose from the collected data obtained from the egocentric sensor, such as CVRA, FPHA (First-Person Hand Action) datasets. HPN, V2V, PtoP are the deep networks/Convolutional Neural Networks (CNNs) for estimating 3D hand pose that uses the point cloud data of the hand. We evaluate the estimation results using the pre-processing step and do not use the pre-processing step to see the effectiveness of the proposed method. In particular, we evaluated the entire FPHA dataset with five different sampling configurations. The results show that 3D distance error is increased many times compared to estimates on the hand datasets are not obstructed (the hand data obtained from surveillance cameras, are viewed from top view, front view, sides view) such as MSRA, NYU, ICVL datasets. The results are quantified, analyzed, shown on the point cloud data of CVAR dataset and projected on the color image of FPHA dataset.**

Index Terms: *3D Hand Pose Estimation, Point Cloud Data, 3D Convolutional Neural Networks, Egocentric Vision Dataset.*

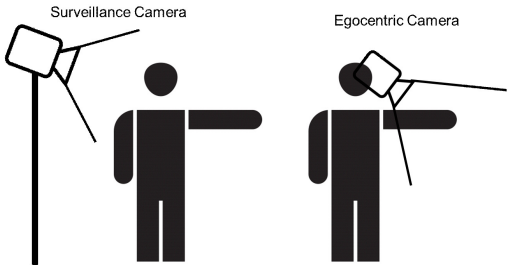

Figure 1: Illustrating the distinction between surveillance cameras and egocentric cameras.

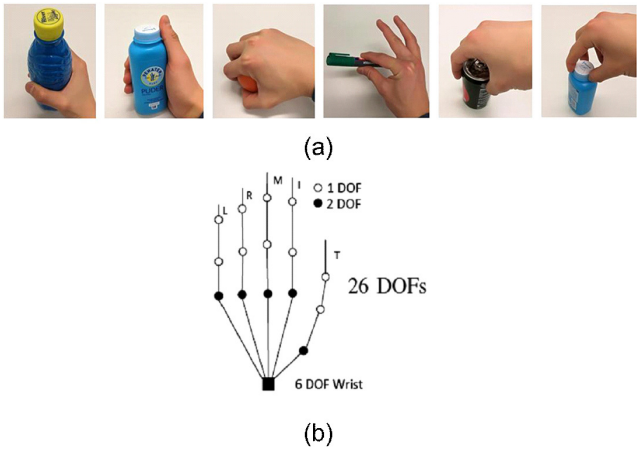

Figure 2: Illustration of some types of grasping the object and the number of Degrees Of Freedom (DOF) of the hand [1].

I. INTRODUCTION

Building robotic arms to perform complex actions like human hands is challenging research. Due to the joints on the human hand, there is a great Degree Of Freedom (DOF), as illustrated in Figure 2(b). In which the re-modeling of human hand actions is an approach [2]. From there build the action of fingers on robot arms like that, as simulating the human hand grasping objects in everyday life [3] (Figure 2(a)). To do this, hand poses in the 3D space / the real world need to be estimated.

The camera mounting is called ”Egocentric Camera” and the dataset obtained from this camera is called ”Egocentric Dataset”. Figure 1 shows the distinction between surveil-

Figure 3: Illustration of the RGB images captured from the mounted camera on the chest's person/ egocentric camera of FPHA dataset.

lance cameras and First-Person/Egocentric cameras. At the same time they are also defined in [4], [5]. The human hand in a real environment from an egocentric vision is missing due to the obscure data of the fingers, usually only obtained data of the palm as Figure 3. The human hand must first be detected, segmented, and fully estimated for joints in the 3D space. However, this issue is faced with many difficulties due to the data of the human hand is lost, obscured, and noises when collected from the egocentric vision.

Over the past four years, with the success of Convolutional Neural Networks (CNNs) in various computer vision, there have been about 60 valuable studies on estimating 3D hand pose using this method and the statistics of studies are shown in the first table of research by Le et al. [6]. The majority of studies on 3D hand pose estimation were evaluated on three outstanding datasets: MSRA [7], NYU [8], ICVL [9]. These datasets have the hand data that is sufficient (not obscured), thus the estimating results of the hand pose are low error, the results are shown in [6]. Recently, several CNNs have been proposed for 3D hand pose estimation for egocentric datasets on the RGB and based on the 3D joints annotation, their estimated results have low errors, as HOPE-Net [10]. The average error estimated HOPE-Net's hand pose on the FPHA [11] dataset was 8.61mm. This dataset have not been evaluated and compared with 3D CNNs appeared earlier as HPN [12], V2V [13], PtoP [14]. HPN, V2V, PtoP are the deep networks that the estimated results are high accuracy (the average of 3D joint error is 10.5 mm, 7.49mm, 7.71mm on the NYU dataset) on the datasets that captured the surveillance camera at the front view or top view. These CNNs estimate the joints of the hand based on point cloud data, it is the same as the real world. Currently, this data is collected from various types of depth sensors such as MS Kinect, Real scene, Creative Interactive Gesture.

The main contributions in our paper are as follows:

- An extension from the previous work [15] on 3D hand pose estimation are proposed. In this study, we deploy an end-to-end frame-work forfor 3D hand pose estimation from complex scene data that are obtained from the egocentric sensor. The proposed frame-work is a combination of the pre-processing step for the separating hand region in the complex scenes and a 3D CNN estimating a hand pose from the point cloud data.
- To adapt challenges when estimating hand poses from missing, obscured data, especially estimating the joints of the hands in the three dimension space,we performed the fine-tuning HPN, V2V, PtoP on the egocentric datasets such as CVAR [16], FPHA [11]. These CNNs achieve good estimation results because the training features are extracted from the point cloud data (3D data) of hand. In [15] we only evaluated performnaces of hand-pose estimation with the CVAR [16]. In this study, we expand the evaluations to the FPHA dataset. It is a large dataset with more complexity. Particularly, we evaluated multiple configuration schemes for training and testing samples.
- The evaluations are performed and shown on 3-D point cloud data. This type of data are more precisely and closed tothe real world expressions. In relevant works (e.g., in Doosti's study [10]), the estimation results are presented in color images. These presentations do not present intuitively poses and limit observing hand/joins in occlusion cases.

The paper is organized as follows: Section I introduces 3D hand pose estimation methods and the application. Section II discusses the related work by the methods, results, sensors, and the egocentric datasets. Section III presents 3D hand pose estimation by 3D CNNs. Section IV presents, discusses the experimental results of 3D hand pose estimation. Section V discusses the conclusion and future work of this paper.

II. Related works

The studies of estimating, restoring the full 3D hand skeleton, and pose involve many applications, the studies are heavily listed in the survey of Rui et al. [17]. This is a very comprehensive listing of 3D hand pose estimation and related studies: hand detection, hand tracking, hand parsing, fingertip detection, hand contour estimation, hand segmentation, gesture recognition. The challenges of estimating hand pose are also presented in the study of Rui et al. [17]: low resolution, self-similarity, occlusion, incomplete data, annotation difficulties, hand segmentation, real-time performance. 3D hand pose estimation is usually based on depth data obtained from depth sensors. The first table of research by Li et al. [17] lists 19 popular depth sensors that can collect depth data produced between 2010 and 2018 and

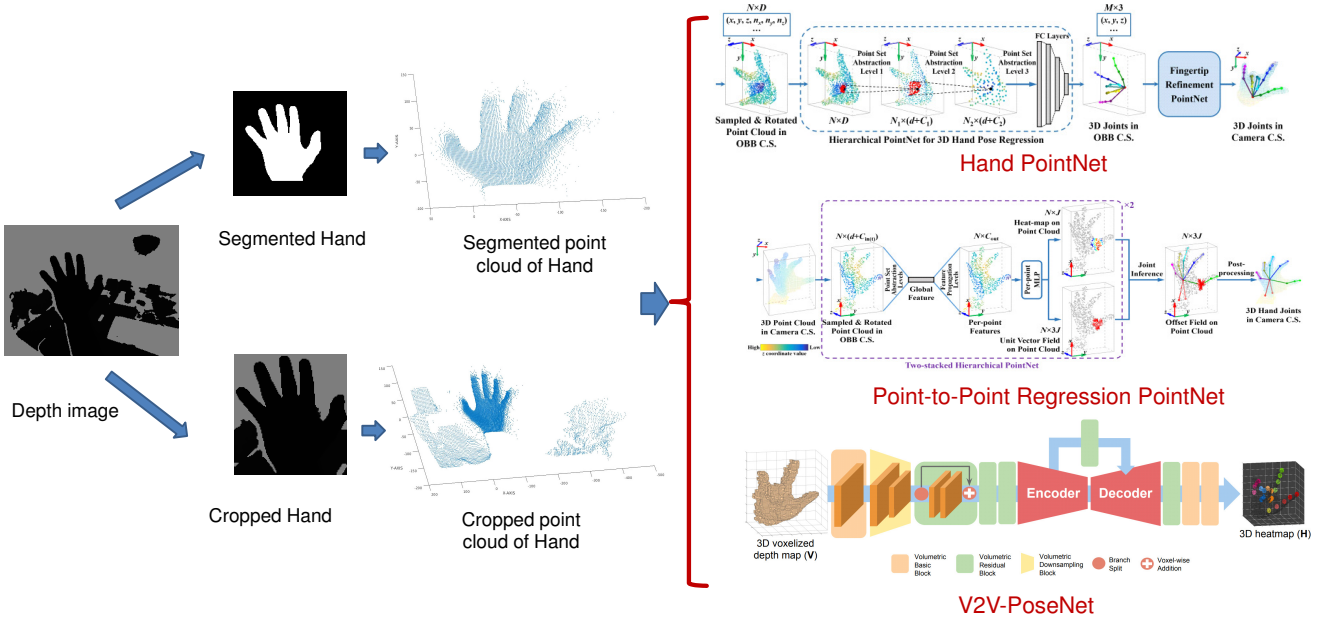

Figure 4: Overview of 3D hand pose estimation approach using 3D CNN-based from the point cloud data.

depth technology, measurement range, the maximum speed of depth data collection are also presented.

The depth sensors are divided into groups.In particle, the weight, the range limitation, and frame rate of the sensors have been improved. The weight, range limitation, and frame rate of MS Kinect v1 (2010) are about 907 grams, 0.5–4.5m, 30fps, respectively; The Intel real scene D435 (2018) are about 150 grams, 0.11–10 m, 90fps, respectively [17]. From this, the quality of the data collected is better.

For the hand pose estimation, Rui et al. [17] has presented three methods to solve 3D hand pose estimation: model-based method, appearance-based method, hybrid method. In which Erol et al. [18] also introduced two main methods (model-based method, appearance-based method) to solve this problem in the 2D space. The model-based method compares the hypothetical hand poses and the data obtained from the cameras. The comparison is evaluated based on objective function measures the discrepancy between the actual observations and the observations that are expected from the generative hand model. The appearance-based method based on learning the characteristics from the observations to a discrete set of the annotated hand poses. The model of training the annotated hand poses is the ability of the discriminative classifier or regression model to describe invariant characteristics of hand pose as a map of the joints of the fingers.

Estimating 3D hand pose has been interested in research with a large community in recent years, especially the method using CNNs. Doosti et al. [19] has been presented, there are two methods for estimating 3D hand pose: depth-based method and RGB-based method. In the depth-based method divided into two branches is to estimate the joints on the depth map of the image depth as researchers of Oberweger et al. [20], Zhang et al. [21], Wan et al. [22]; converting the depth image data to the point cloud data and estimate on it as researchers of Liuhao et al [23], Wan et al. [24], Ge et al. [25]. The RGB-based method trains the position of the annotated joints on the color image, then predicts the positions of the joints on the 2D heat-map. Finally, the regression of the 3D position of the joints is based on a synthesized of joints data or projected to 3D space through the depth image and point cloud data, respectively. Recently, the study of Doosti et al. [10] has very good results of 3D hand pose estimation on FPHA dataset (the average error is 6.81mm). This study used the ResNet10 to predict the initial 2D coordinates of the joints, after that using the graph convolution to estimate the better 2D pose. And then the predicted joints are passed to Graph U-Net [26] to find the 3D hand joints.

To evaluate the hand pose estimation methods, Rui et al. [17] proposed an evaluation experiment as shown in 4^{th} figure of research by Li et al. [17]. In particular, the benchmark datasets for evaluating 3D hand pose estimation were also presented: NYU [8], ICVL [27], and MSRA [28].

Another good survey of hand pose estimation is the study of Barsoum [29]. In this study, the author also conducted surveys on three methods to perform hand pose estimation from the depth image: appearance-based method is shown

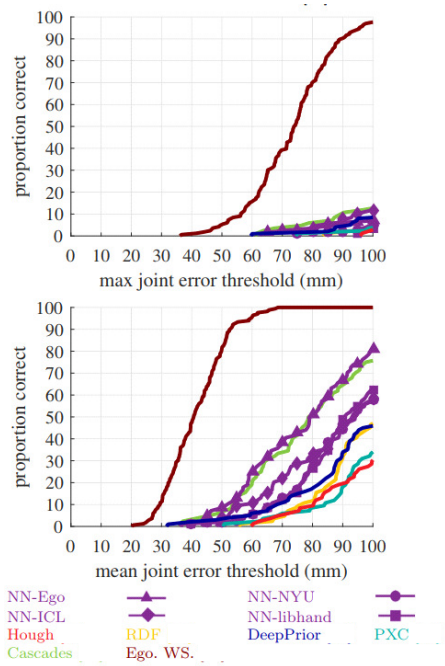

Figure 5: Distribution of 3D joints error estimation [36] on the UCI-EGO-Syn [37] dataset.

in 3^{th} figure, model-based method is shown in 4^{th} figure, the hybrid method is shown in 4^{th} figure of research by Barsoum et al. [29]. Hand segmentation steps are included in the three models because its outcome determines the accuracy of all these steps. This step can be solved by several methods as follows: Color or IR skin based; Temperature-based; Marker-based; Depth-based; Machine learning-based. In particular, Barsoum et al. [29] presented two studies that use deep learning to estimate hand pose in 2014 [8] and 2015 [30].

By egocentric dataset, already has a number of datasets published, these are listed as follows: UCI-EGO dataset [31], Graz16 dataset [16], Dexter+Object dataset [32], BigHand2.2M dataset [33], HO-3D Dataset [34]. The results of studies on estimating 3D hand pose before 2019 have great joints error, as illustrated in Figure 5. Recently, Doosti et al. [10] proposed and evaluated Hope-Net on FPHA [35] and HO-3D [34] datasets, the average 3D estimated joints error is very small (6.81mm).

III. 3D HAND POSE ESTIMATION FROM EGOCENTRIC VISION DATASETS

Different from previous studies by using CNNs for 3D hand pose estimation, they are evaluated on MSRA, NYU, ICVL datasets. These datasets are detected and segmented the human hands, as illustrated in Figur 6. In this paper, we have fine-tuning the HPN, V2V, PtoP to estimate 3D hand pose on the CVAR [16], FPHA [11] datasets. Before retrain the estimation model for 3D hand pose estimation on the CVAR dataset, as illustrated in Figure 4. We propose a simple pre-processing step to segment the hand data in the complex scenes. At the same time, we also re-trained and evaluated 3D hand pose estimation on the FPHA dataset that based on the 3D annotation of hand joints.

1. Hand segmentation from egocentric vision data

Previous studies on 3D CNNs estimated the hand pose on the hand data which have been segmented with data of the complex scenes. Ge et al. [14] used a single hourglass network [38] to detect 2D hand joint locations on the 2D heat-map of RGB image and use the corresponding depth information for hand segmentation. Ge et al. [12] used the random decision forest (RDF) [9] to segment the data of hand on the depth image. The presented methods of hand segmentation with the complex scene are complex, time consuming for feature extraction, hand model training.

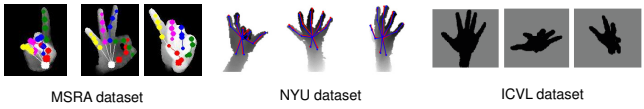

Figure 6: Illustration of the detected and segmented human hands on the MSRA, NYU, ICVL datasets.

Based on the contextual, as illustrated in Figure 3. The data of hand and the complex scene are collected from the egocentric vision sensor. Before performing 3D hand pose estimation from the point cloud data, we propose a pre-processing step to segment the hand from the complex scene data, as shown in Figure 4. The depth image contains the depth value of the hand data is the closest (the hand is closest to the sensor). From there we use a threshold d_{thres} which is the maximum depth value of the hand to segment the data of the hand and other data in the complex scene. The data segmentation algorithm on depth image obtained from egocentric vision is presented in Algorithm 1.

2. 3D hand pose estimation by Point Cloud-based

Different from previous CNN 3D hand pose estimation methods that take 2D images [22] or 3D volumes [25] as the input, the HPN, V2V, PtoP processes directly on the point cloud data generated from the depth images. The point cloud data of the segmented hand is represented by the Kd-tree [39] structure when used the HPN, V2V, PtoP on the MSRA, NYU, ICVL datasets.

a) HPN for 3D hand pose estimation

From this structure, it is possible to exploit the feature of the normal vector of points on the point cloud data. Due to a large number of points on the point cloud data and

Algorithm 1: The data segmentation algorithm on the depth image obtained from egocentric vision, based on the threshold.

Data: Depth image I_d
Result: Depth image I_h

$w = width(I_d)$;
$h = height(I_d)$;
$I_h = zeros(w, h)$;
for i=1 to w
  for j=1 to h
    $D_{value} = I_d(i, j)$;
    If($D_{value} > 0$ && $D_{value} < d_{thres}$)
      $I_h(i, j) = D_{value}$;
    else
      $I_h(i, j) = 0$;
    endif
  endfor
endfor

the 3D space, to reduce the volume of calculations and to make the network is robust, HPN has implemented the data standardization and rotation with the 3D point cloud in an Oriented Bounding Box (OBB) and data reduction in three levels, as illustrated in Figure 4(top branch): Level 1 data is standardized to 1024 points; Level 2 includes 512 points; Level 3 consists of 128 points. This is a simple pre-processing step, but it is highly effective for calculating time.

OBB is a tightly fitting bounding box of the input point cloud. The orientation of OBB is determined by performing the Principal Component Analysis (PCA) on the input point cloud. The x, y, z axes of the OBB coordinate system are determined by the eigenvectors of input points' covariance matrix, which correspond to eigenvalues from largest to smallest, respectively. The input point cloud p^{ori} is first transformed into OBB space, then these points are shifted to have zero mean and scaled to a unit size as follows Eq. (1).

$$p^{obb} = (R_{obb}^{ori})^T . p^{ori}, p^{nor} = (p^{obb} - \overline{p}^{obb})/L_{obb} \quad (1)$$

where R_{obb}^{ori} is the rotation matrix of the OBB in camera coordinate system; p^{ori} and p^{obb} are 3D coordinates of point p in camera coordinate system and OBB coordinate system, respectively; $\overline{p}^{obb}$ is the centroid of point cloud $\{p_i^{obb}\}_{i=1}^N$; L_{obb} is the maximum edge length of OBB; p^{nor} is the normalized 3D coordinate of point p in the normalized OBB coordinate system.

HPN is based on the PointNet [40], [41], it exploits the hierarchical PointNet for 3D hand pose estimation. The input of HPN is a set of normalized points PO, as presented in Eq. (2).

$$PO = \{po_i\}_{i=1}^N = \{p_i^{nor}, n_i^{nor}\}_{i=1}^N \quad (2)$$

where p_i is the 3D coordinate of the normalized point and n_i^{nor} is the corresponding 3D normal vector of p_i; N is the number of points of the standardized hand point cloud data then fed into a hierarchical PointNet [40].

The first two levels of group input points, as shown in Figure 4 (top branch), are $N_1 = 512$ and $N_2 = 128$ local regions, respectively and extract $C_1 = 128$ and $C_2 = 256$ dimensional features for each local region, respectively. Each local region contains k = 64 points. The last level extracts a 1024-dim global feature vector which is mapped to an F-dim output vector by three fully-connected layers. The PointNet is designed to output an F-dim $(F < 3 \times M)$ representation of the estimated 3D hand pose.

In the training process, HPN given the training samples with the normalized point cloud and the corresponding ground truth 3D joint locations $(po_T^{nor}, gr_T^{nor})_{t=1}^T$. It minimizes the following objective function, as shown in Eq. (3).

$$\theta^* = arg\min_{\theta} \sum_{t=1}^{T} \|\alpha_t - \gamma(po_T^{nor}, \theta)\|^2 + \lambda \|\theta\|^2 \quad (3)$$

where θ is the network parameters; γ represents the hand pose regression PointNet; λ is the regularization strength; α_t is an F-dim projection of gr_T^{nor}. By performing PCA on the ground truth 3D joint locations in the training dataset, the network can obtain $\alpha_t = E^T.(gr_T^{nor} - u)$, where E is the principal component, and u is the empirical mean.

In the testing process, the estimated 3D joint locations are reconstructed from the HPN outputs, as shown in Eq. (4).

$$\widehat{gr}^{nor} = E.\gamma(po^{nor}, \theta^*) + u \quad (4)$$

b) PtoP for 3D hand pose estimation

Just like the two methods presented, the input to estimate PtoP's 3D hand pose is also the point cloud data. To predict the hand poses that use the regression approach they are often require learn a highly non-linear mapping. However, if learning on all points in the point set is unnecessary and may make the per-point votes noisy, especially the training time is high. The main contributions of PtoP are to generate point-wise estimations of hand joint locations from the point cloud, which is able to better utilize the local evidence. The point-wise estimations can be defined as the offsets from points to hand joint locations. The process of estimating the offset of the hand joints is illustrated

in 2^{th} figure of research by Ge et al. [14]. PtoP uses the hierarchical PointNet [40] for learning heat-maps and unit vector fields on 3D point cloud. They are defined in Eq. (5) and Eq. (6), respectively.

$$H(p_i, \phi_j^*) = \begin{cases} 1 - ||\phi_j^* - p_i||/r & p_i \in \phi_j^* \\ and \quad ||\phi_j^* - p_i|| \leq r, \\ 0 \quad otherwise; \end{cases} \tag{5}$$

$$U(p_i, \phi_j^*) = \begin{cases} (\phi_j^* - p_i)/||\phi_j^* - p_i|| & p_i \in \phi_j^* \\ and \quad ||\phi_j^* - p_i|| \leq r, \\ 0 \quad otherwise; \end{cases} \tag{6}$$

where H is the heat-map of the neighboring points with the ground truth hand joint location; U is unit vector field/normal vector of the neighboring points with the ground truth hand joint location; $\phi_j^*(j = 1, ..., J)$ is is a set of K-Nearest Neighboring points (KNN) of the ground truth hand joint location; $p_i(i = 1, ..., N)$ is a set of points that want to estimate the offset; r is the maximum radius of ball for nearest neighbor search.

To get standard data as input to CNN, this method also performs the data normalization process to unify the view direction and size of the data. PtoP creates an Oriented Bounding Box (OBB) from the 3D point cloud and transform the 3D points into the OBB coordinate system, as shown in Figure 4. Standardized point cloud data is between -0.5 and 0.5 and includes 1024 points. The network architecture of PtoP uses a single hierarchical PointNet [40]. Base on the stacked hourglass networks method, PtoP uses two-stacked hierarchical PointNet modules end-to-end to boost the performance of the network with the same hyper-parameters. The architecture of the two stacks is shown in 4^{th} figure of research by Ge et al. [14]. To supervise the training process of the two-stacked hierarchical PointNet, PtoP uses the loss function which is defined in 4^{th} equation of research by Ge et al. [14].

c) V2V for 3D hand pose estimation

As shown in Figure 4 (bottom branch), the input of V2V method is 3D voxelized data. Thus, it reprojects each pixel of the depth map to the 3D space. After reprojecting all depth pixels, the 3D space is discretized based on the pre-defined voxel size. V2V-PoseNet is based on the hourglass model [38] and is designed to be divided into four volumetric blocks. The first volumetric basic block includes a volumetric convolution, volumetric batch normalization, and the activation function. This block is located in the first and last parts of the network. The second volumetric residual block extended from the 2D residual block in [42]. The third volumetric downsampling block that is identical to a volumetric max pooling layer. The last block is the volumetric upsampling block, which consists of a volumetric deconvolution layer, volumetric batch normalization layer, and the activation function.

Each phase of V2V-PoseNet method presented in Figure 4. Each phase consists of four blocks: Volumetric residual block, Volumetric downsampling block, Branch split, Voxel-wise addition. Therein, the volumetric downsampling block reduces the spatial size of the feature map while the volumetric residual bock increases the number of channels in the encoder phase. Otherwise, the volumetric upsampling block enlarges the spatial size of the feature map. When upsampling, the network decreases the number of channels to compress the extracted features. To predict each keypoint of the hand in 3D space through two stages: encoder, decoder. They are connected together by the voxel-wise. To supervise the per-voxel likelihood in the estimating process, V2V generates 3D heat-map, wherein the mean of Gaussian peak is positioned at the ground truth joint location in Eq. 7.

$$D_n^*(i, j, k) = exp\left(- \frac{(i - i_n)^2 + (j - j_n)^2 + (k - k_n)^2}{2\sigma^2} \right) \tag{7}$$

where D_n^* is the ground truth 3D heatmap of n^{th} keypoint, (i_n, j_n, k_n) is the ground truth voxel coordinate of n^{th}, and $\sigma = 1.7$ is the standard deviation of the Gaussian peak [13]. V2V also uses the mean square error as a loss function L in Eq. 8.

$$L = \sum_{n=1}^{N} \sum_{i,j,k} ||D_n^*(i, j, k) - D_n(i, j, k)||^2 \tag{8}$$

where D_n^* and D_n are the ground truth and estimated heatmaps for n^{th} keypoint, respectively, and N denotes the number of keypoints.

IV. Experimental Results

1. Datasets

In this paper, we conduct fine-tuning and evaluation on two egocentric vision datasets as following.

CVAR dataset [16] has more than 2000 depth frames of several egocentric sequences of six subjects. It captured from Creative Senz 3D, with intrinsics $fx = 224.5, fy = 230.5, ux = 160, uy = 120$ and contains several sequences of a single person. 3D annotations are made with 21 joint points, as illustrated in Figure 9. The size of the image is 320×240 pixels. The authors proposed a semi-automated the application that makes it easy to annotate sequences of articulated poses in the 3D space. This application asks

a human annotator to provide an estimate of the 2D re-projection of the visible joints in frames they are called reference frames. It proposes a method to automatically select these reference frames to minimize the annotation effort, based on the appearances of the frames over the whole sequence. It then uses this information to automatically infer the 3D locations of the joints for all the frames, by exploiting appearance, temporal and distances constraints. Figure 7 (top row) presents the depth images, the corresponding segmented human hands (bottom row), respectively.

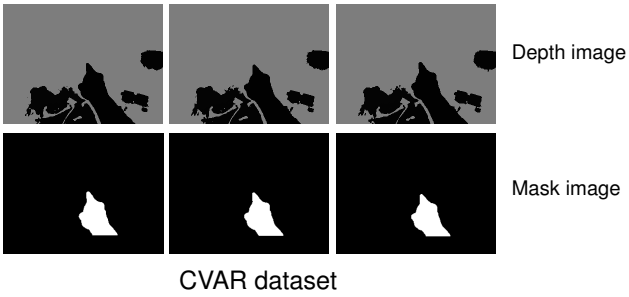

Figure 7: Illustration of CVAR [16] dataset.

FPHA dataset (First-Person Hand Action Dataset) [35] includes 100K frames, involving 26 different objects in several hand configurations. This dataset is collected from 6 people (6 subjects), 45 daily hand action categories, each action is collected from 3 to 9 times (instances), for a total of 1175 action sequences. It is captured from the Intel RealSense SR300, the size of color data is 1920x1080 pixels with *.jpeg* format and the size of depth data is 640x480 pixels with *.png* format, as illustrated in Figure 8. It also provided 3D hand joints annotation by the MOCAP system with 21 joints, as illustrated in Figure 9.

Figure 8: Illustration of FPHA [35] dataset. The MOCAP system used to make 3D hand joints annotation.

In this paper, to see the effect of the pre-processing step before performing 3D hand pose estimation. We evaluated the hand pose estimation on the segmented and cropped hand/no segmented hand data in the complex scenes of the CVAR dataset. On the FPHA dataset, the hand holds the object so the hand and object data are stuck together. Therefore, we perform fine-tuning and estimation on the cropped hand data based on the annotation data of the hand. This data is the same as the non-segmented hand data on the CVAR dataset.

2. CNNs parameters and evaluation measurement

We perform experiments on PC with Core i5 processor - RAM 8G, 4GB GPU. Pre-processing steps were performed on Matlab, fine-tuning, and development process in Python language on Ubuntu 18.04.

Hand PointNet parameters:

For training PointNets, we use Adam [43] optimizer with initial learning rate 0.001, batch size 8, and regularization strength 0.0005. The learning rate is divided by 10 after 50 epochs. The training process is stopped after 60 epochs. The number of PCA components is 42, the size of KNN search is 64. The square of radius for ball query in level 1, level 2 is 0.015, 0.04, respectively. This set of parameters is the same as in the experiment of [12]. Implementation details of HPN are shown in the link [1].

V2V-PoseNet parameters:

This deep network is developed in the PyTorch framework. The zero-mean Gaussian distribution with $\sigma = 0.001$ is initialized to all weights. The learning rate is set 0.00025 and batch size is set 4. The size of the input to the proposed system is $88 \times 88 \times 88$. This deep network also uses the optimizer method of Adam [43]. To standardize data for training, V2V rotates [-40, 40] degrees in XY space, scale [0.8, 1.2] in 3D space, and translate with the size of voxels [-8, 8] in 3D space. We trained the model for 15 epochs. Implementation details of V2V are shown in the link [2].

Point-to-Point Regression PointNet parameters:

The deep neural network models are implemented within the PyTorch framework. This deep network also uses the optimizer method of Adam [43], the learning rate 0.001, batch size 8, momentum 0.5 and weight decay 0.0005. The learning rate is divided by 10 after 30 epochs. The training process is stopped after 60 epochs. Implementation details of PtoP are shown in the link [3].

Evaluation measure:

Like the evaluations of the previous 3D hand pose estimation method, we used the average 3D distance error (as shown in Eq. 9) to evaluate the results of the 3D hand pose estimation on the datasets.

$$\widehat{Err_a} = \frac{1}{Num_s} \sum_{n=1}^{Num_s} \frac{1}{21} \sum_{k=1}^{21} DIS(p_g, p_e) \qquad (9)$$

[1] https://github.com/3huo/Hand-Pointnet.
[2] https://github.com/dragonbook/V2V-PoseNet-pytorch.
[3] https://github.com/ataata107/Point-to-Point-Regression-Hand-Pose.

where $DIS(p_g, p_e)$ is the distance between a ground truth joint p_g and an estimated joint p_e; Num_s is the number of testing frames. In this paper, we evaluated the 21 joints of hand pose, illustrated in Figure 9.

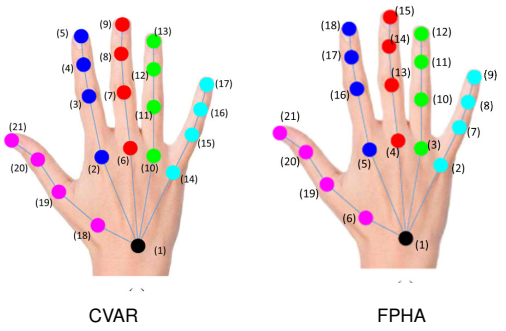

Figure 9: Illustrating the hand joints of CVAR and FPHA dataset.

In this paper, we use the rate at approximately 1:4 for testing and training samples with the CVAR dataset. This ratio is based on the setup given in [16]. This means that on the CVAR dataset using 1733 samples for training and 433 samples for testing. For each trial, the training and testing samples are selected randomly.

On the FPHA dataset, we used five configurations for training and testing. The first configuration used "Subject2", "Subject4", "Subject5", "Subject6" with 72260 samples for training and "Subject3" with 15770 samples for testing (the ratio is at approximately 1: 4 for testing and training model).

The second configuration ($Configu\_312$) used the 3^{rd} sequence in each subject "Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (23704 samples), the 1st sequence in each subject for validation (27097 samples) and remaining for training (54658 samples) (the ratio is at approximately 1:3 for testing and training model). The selection of samples was based on HOPE-Net [10] evaluations on the FPHA dataset.

The third configuration ($Configu\_123$) used the 1^{st} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (27097 samples), the 2^{nd} sequence in each subject for validation (25475 samples) and remaining for training (52887 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

The fourth configuration ($Configu\_213$) used the 2^{nd} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (25475 samples), the 1^{st} sequence in each subject for validation (27097 samples) and remaining for training (52887 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

The fifth configuration ($Configu\_321$) used the 3^{rd} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (23704 samples), the 2^{nd} sequence in each subject for validation (25475 samples) and remaining for training (56280 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

In this paper, we have evaluated with many different sampling configurations training and testing 3D hand pose estimation models. From the second configuration to the fifth configuration (this experiment is based on the multi-fold not just n-fold cross-validation), we only evaluate the **cropped hand** method.

3. Results and Discussions

Like the evaluation of 3D hand pose estimation results shown in the 2^{nd} table of research by Yoo et al. [44], we also use the 3D distance error (mm) to evaluate the estimation results on CVAR and FPHA datasets with the five configurations. The average 3D distance error and the average processing time on the first configuration are shown in Table I, II, respectively.

The processing time of 3D hand pose estimation is shown in Table II. This time includes the hand segmentation time or the cropping hand time on the depth image. The distribution of the average 3D distance error when evaluating on CVAR dataset is presented in Figure 10.

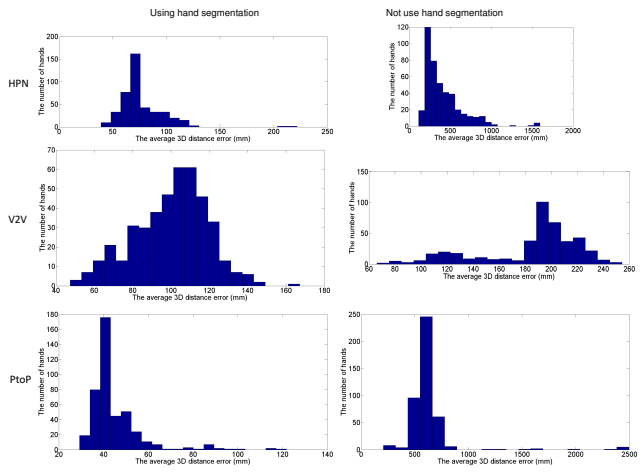

Figure 10: The distribution of average 3D distance error for 3D hand pose estimation on the CVAR [16] dataset when using the hand segmentation and not use the hand segmentation. The left is the error distribution of estimating 3D hand pose from the point cloud data using the hand segmentation. The right is the error distribution of estimating 3D hand pose from the point cloud data which not use the hand segmentation. The top line is based on HPN method; The mid line is based on V2V method; The bottom line is based on PtoP method.

TABLE I: The average 3D distance error (mm) of HPN [12], V2V [13], PtoP [14] on the CVAR, FPHA (the first configuration) datasets for 3D hand pose estimation when using the hand segmentation and the cropped hand.

Dataset	**Measurement/ Method**		**HPN [12]**	**V2V [13]**	**PtoP [14]**
CVAR	**Average of 3D joints error (mm)**	**Hand segmentation**	**77.53**	**100.58**	**45.27**
		Cropped hand	450.87	184.88	617.99
FPHA		**Cropped hand**	144.16	241.23	121.74

TABLE II: The average processing time (mm) of HPN [12], V2V [13], PtoP [14] to estimate a 3D hand pose on the CVAR dataset when using the hand segmentation and the cropped hand.

Measurement/ Method		**HPN [12]**	**V2V [13]**	**PtoP [14]**
Processing time (ms)/ hand	**Hand Segmentation**	**12.44**	**504.22**	**197.81**
	Cropped hand	17.26	638.25	198.52

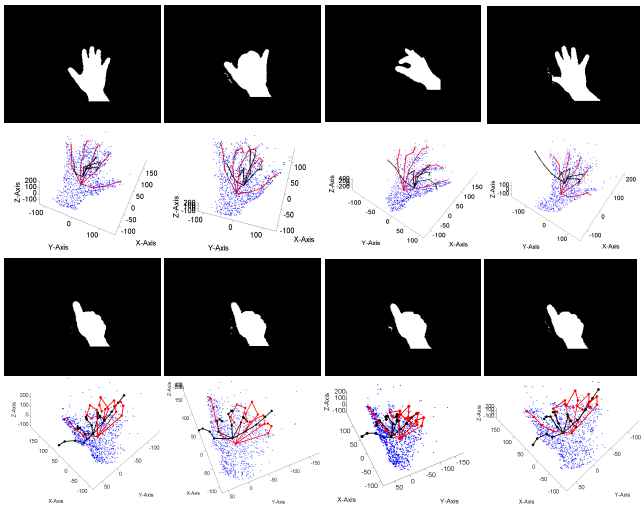

Figure 11: 3D hand pose estimation results on the CVAR [16] dataset; Top line is the depth images; The bottom line is 3D hand pose estimation results. The blue point cloud is the normalized point cloud of the hand. The red hand skeleton is 3D ground truth of 3D hand pose. The black hand skeleton is the estimated 3D hand pose.

Based on the results in Table I and Figure 10 when using the hand segmentation, the estimated error of joints in the 3D space on the CVAR dataset is more than 9 times higher than the errors on the MSRA dataset, more than 7 times the error on the NYU dataset, 11 times the error on the ICVL dataset, respectively. This problem starts from the complexity of the CVAR dataset, the data of the hand is obscured, is missing because the view direction of the sensor only sees the palm, the data of the fingers is obscured, as shown in Figure 11. Another problem is that the depth value of the hand data on the CVAR dataset is larger than the MSRA, NYU, ICVL. For example, the minimum hand depth value of 10 frames on CVAR is about 510mm while the minimum hand depth value of 10 frames

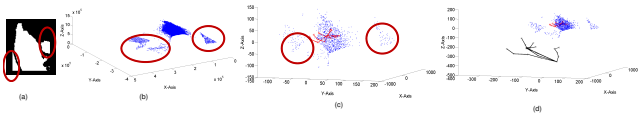

Figure 12: Illustration of estimation result on the hand data that cropped/did not segmented in the complex scene. (a) is the depth image data of the hand cut based on the ground truth data; (b) is point cloud data that has not been sample reduced; (c) is the point cloud data that is sampled and normalized, the ground truth skeleton/joints data; (d) is 3D hand pose estimation results based on standardized data based on HPN (black skeleton). The data areas marked with red circles are data of other objects in the complex scene.

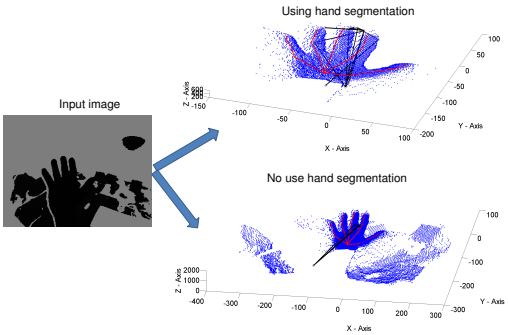

Figure 13: Illustration of 3D hand pose estimation result on the hand data when using hand segmentation and the cropped hand/not use hand segmentation in the complex scene that base on the V2V method. The blue points are the point cloud data of the hand without sample reduction and data normalization. The red skeleton is 3D ground truth of hand pose, the black skeleton is 3D hand pose estimation.

on MSRA is about 350mm. This means that the hand on the CVAR dataset is far more sensor than the MSRA dataset.

Figure 11 illustrates the result of estimating 3D hand pose on the obscured data by these fingers in the CVAR dataset. Although on the segmented hand there is little data noise, as illustrated in Figure 11. However, this data is few and not far from the hand data, so the estimation range of the hand pose is not much larger, so the estimation results when using the hand data segmentation is better more than in the case of not using the hand data segmentation.

When the hand data is not segmented in a complex scene the error is greatly increased (from 77.53 to 450.87 on the CVAR dataset), as shown in Table I and illustrated in Figure 12. This result is greatly increased because the estimated space for 3D hand joints on the non-segmefnted data is very large and there is a high dimension in the 3D space.

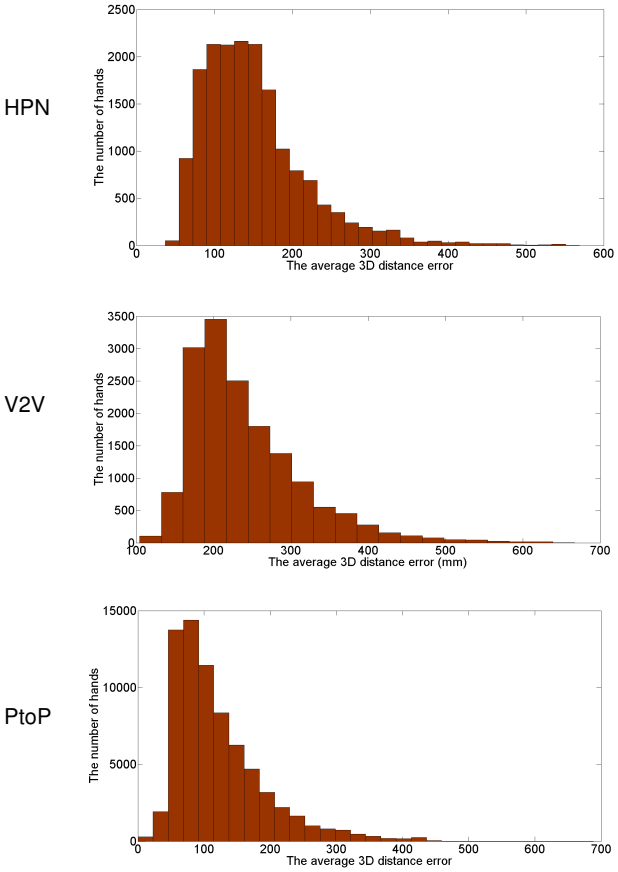

Figure 14: The distribution of average 3D distance error for 3D hand pose estimation on the FPHA [35] dataset when not use the hand segmentation. The top line is based on HPN method; The mid line is based on V2V method; The bottom line is based on PtoP method.

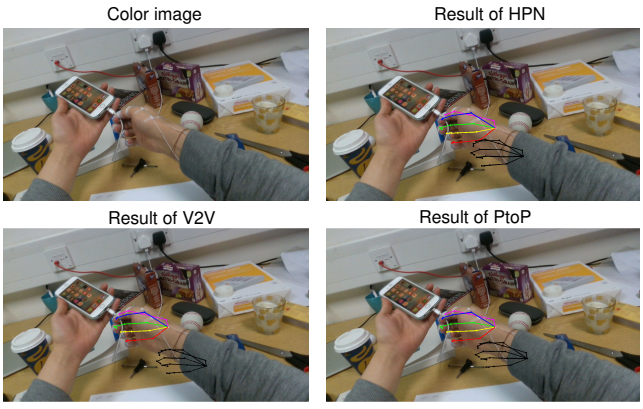

Figure 15: The estimated results on color image projected from 3D hand pose data by HPN, V2V, PtoP on the FPHA [35] dataset. 3D ground truth hand data with fingers magenta, blue, green, yellow, red. The estimated 3D pose of the hand is black.

The estimated 3D hand joints error and error distribution on the FPHA dataset with the first configuration are shown in Table I and Figure 14, respectively. The estimated 3D hand joints error is from 144.16 to 241.23mm.

The second configuration to fifth configuration results on the testing sets is shown in Table III. Although the estimated error was much lower than the results in the first configuration. However, their results are many times higher than the estimated results using HOPE-Net (6.81mm). The estimated results improved due to the posture learning of 6 people performing the activities. In the first configuration, only five people can learn the activities. Each person has a different hand size, while performing the same activity, each person's style is different. Especially, the training is done on 3D joints data, so it contains many challenges. The results of the FPHA dataset were evaluated on five configurations, in HOPE-Net only evaluated on a configuration similar to $Configu\_312$, but only evaluated on about 1000 samples. In this paper, we evaluate the entire dataset.

TABLE III: The average 3D distance error (mm) of HPN [12], V2V [13], PtoP [14] on the FPHA dataset with $Configu\_312$, $Configu\_123$, $Configu\_213$, $Configu\_321$ for 3D hand pose estimation when using the cropped hand.

Dataset	Measurement	HPN	V2V	PtoP
Configu_312	**Average of 3D joints error (mm)**	135.54	**115.34**	154.33
Configu_123		140.26	**123.85**	160.48
Configu_213		139.42	**112.78**	188.72
Configu_321		147.86	**129.46**	179.82

Figure 16 presented the average 3D distance error (mm) on each epoch of HPN that is based on the second configuration to the fifth configuration. Figure 15 shows the results of 3D hand pose estimation of HPN, V2V, PtoP projected from 3D data to color image.

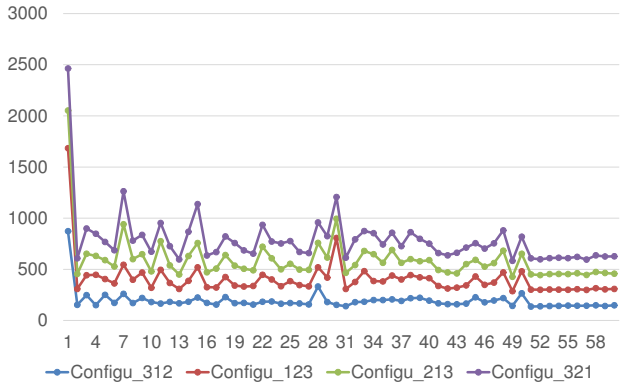

Figure 16: The average 3D distance error (mm) on each epoch of HPN that based on the second configuration to fifth configuration.

All three CNNs (HPN, V2V, PtoP) that we use for training and estimating 3D hand pose are in feature-based methods. Estimation results always depend entirely on the characteristics learned from the datasets. Therefore, the estimation results depend on the features learned from the two datasets (CVAR, FPHA). The fact that FPHA is a dataset which is contructed with many activities in daily life. These activities are perfomed in a natural way. The features arecalculated and extracted based on the point

cloud data (3D data) of the hand data as illustrated in Figure 4. Ultilizing the trained models, one can easily deploy transfer learning for a new hand pose dataset. For instance, another method such as HOPE-Net utilizing trained model from FPHA to detect the key points on an RGB image and then deploy the Adaptive Graph U-Net technique to estimate the detected key pointsinto 3D space. Because it always take time and label costs to prepare ground-truth of 3-D hand pose datasets. In the liteturatue works, there is not so many available datasets which support 3-D hand pose's ground-truth. We will continue consider other 3-D egocentric vision datasets whose ground-truth measurement are available.

V. Conclusion

Estimating the 3D hand pose from image data obtained from egocentric vision is an issue that contains many challenges. Most recent studies of 3D hand pose estimation using CNNs were evaluated on simple datasets such as MSRA, NYU, ICVL. These datasets have the data of hands that are complete. However, in practice developing the human assistance systems, such as assistance systems that grasping objects in complex environments, the sensors are often mounted on the person. So the hand data is often obscured by the objects or only the palm area is visible. To see the challenges in these data for fully estimating/restoring hands pose in the 3D space. We proposed a complete pipeline for estimating 3D hand pose from the complex scene data obtained of the egocentric sensor and performed an experiment using the common CNNs (HPN, V2V, PtoP) to estimate hand pose on the datasets obtained from an egocentric vision sensor (CVAR, FPHA datasets). The result found that the 3D hand pose estimation error is increased many times compared to the estimate on the full hand data datasets when using the hand segment and cropped hand/not using the hand segment. We evaluated the 3D hand pose estimation results across various configurations and all frames of the FPHA dataset. In particular, the results show that when using 3D CNNs to estimate the hand pose on the point cloud data, there is a large mean error (greater than 100mm). The results are displayed on the CVAR dataset, illustrated in the point cloud data and projected onto the color image on the FPHA dataset.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.315.

References

[1] Z. Deng, G. Gao, S. Frintrop, F. Sun, C. Zhang, and J. Zhang, "Attention based visual analysis for fast grasp planning with a multi-fingered robotic hand," *Frontiers in Neurorobotics*, vol. 13, no. July, pp. 1–12, 2019.

[2] M. F. Reis, A. C. Leite, and F. Lizarralde, "Modeling and control of a multifingered robot hand for object grasping and manipulation tasks," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 54rd IEEE. IEEE, 2015, pp. 159–164.

[3] J. A. Corrales, C. A. Jara, and F. Torres, "Modelling and simulation of a multi-fingered robotic hand for grasping tasks," in *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, 2010, pp. 1577–1582.

[4] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 407–414.

[5] S. Mann, K. M. Kitani, Y. J. Lee, M. S. Ryoo, and A. Fathi, "An introduction to the 3rd workshop on egocentric (first-person) vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 827–832.

[6] L. Van-Hung and N. Hung-Cuong, "A survey on 3d hand skeleton and pose estimation by convolutional neural network," *Advances in Science, Technology and Engineering Systems Journal*, no. 7, pp. 144–159, 2020.

[7] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[8] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics*, vol. 33, no. 5, 2014.

[9] D. Tang, H. Chang, A. Tejani, and T. Kim, "Latent regression forest: Structured estimation of 3D hand poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1374–1387, 2017.

[10] B. Doosti, S. Naha, M. Mirbagheri, and D. Crandall, "Hope-net: A graph-based model for hand-object pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[12] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet : 3D Hand Pose Estimation using Point Sets," *Cvpr*, pp. 3–5, 2018. [Online]. Available: http://openaccess.thecvf.com/content-cvpr2018/papers/GeHandPointNet3DCVPR2018paper.pdf

[13] G. Moon, J. Y. Chang, and K. M. Lee, "V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 20118, pp. 5079–5088.

[14] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *European Conference on Computer Vision*, vol. 11217 LNCS, 2018, pp. 489–505.

[15] L. Van-Hung, H. Van-Nam, V. Hai, L. Thi-Lan, T. Thanh-Hai, and V. Viet-Vu, "Using Hand PointNet-based 3D Hand Pose Estimation in Egocentric Datasets," in *2020 International Conference on Advanced Technologies for Communications (ATC)*, 2020, pp. 215–220.

[16] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3D training data for fine hand pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 4957–4965.

[17] Y. Li, Z. Xue, Y. Wang, L. Ge, Z. Ren, and J. Rodriguez, "End-to-End 3D Hand Pose Estimation from Stereo Cameras," in *BMVC*, 2019, pp. 1–13.

[18] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[19] B. Doosti, "Hand Pose Estimation: A Survey," *CoRR*, vol. abs/1903.01013, 2019. [Online]. Available: http://arxiv.org/abs/1903.01013

[20] M. O., P. W., and V. L., "Training a feedback loop for hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015, pp. 3316–3324.

[21] Y. Zhang, C. Xu, and L. Cheng, "Learning to Search on Manifolds for 3D Pose Estimation of Articulated Objects," *CoRR*, vol. abs/1612.00596, 2016. [Online]. Available: http://arxiv.org/abs/1612.00596

[22] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Crossing Nets : Combining GANs and VAEs with a Shared Latent Space for Hand Pose Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D Hand Pose Estimation from Single Depth Images Using Multi-View CNNs," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4422–4436, 2016.

[24] C. Wan, A. Yao, and L. Van Gool, "Hand pose estimation from local surface normals," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, 2016, pp. 554–569.

[25] L. G., H. Liang, J. Yuan, and D. Thalmann, "3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation from Single Depth Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] H. Gao and S. Ji, "Graph u-nets," in *International Conference on Machine Learning*, 2019, pp. 2083–2092.

[27] D. Tang, H. J. Chang, A. Tejani, and T. K. Kim, "Latent regression forest: Structured estimation of 3D hand poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1374–1387, 2017.

[28] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 824–832.

[29] E. Barsoum, "Articulated Hand Pose Estimation Review," *CoRR*, vol. abs/1604.06195, pp. 1–50, 2016. [Online]. Available: http://arxiv.org/abs/1604.06195

[30] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," in *Computer Vision Winter Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1502.06807

[31] G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan, "3d hand pose detection in egocentric rgb-d images," in *European Conference on Computer Vision*. Springer, 2014, pp. 356–371.

[32] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *Proceedings of European Conference on Computer Vision (ECCV)*, October 2016. [Online]. Available: http://handtracker.mpi-inf.mpg.de/projects/RealtimeHO/

[33] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T. K. Kim, "BigHand2.2M benchmark: Hand pose dataset and state of the art analysis," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 2605–2613.

[34] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3196–3206.

[35] G. Garcia-Hernando, S. Yuan, S. Baek, and T. K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 409–419.

[36] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: methods, data, and challenges," *International Journal of Computer Vision*, vol. 126, no. 11, pp. 1180–1198, 2018.

[37] G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan, "3d hand pose detection in egocentric rgb-d images," in *European Conference on Computer Vision*. Springer, 2014, pp. 356–371.

[38] N. A., Y. K., and D. J., "Stacked hourglass networks for human pose estimation," in *In European Conference on Computer Vision*, 2016.

[39] Y. Aggarwal, "K Dimensional Tree (K D Tree)," https://iq.opengenus.org/k-dimensional-tree/, [Accessed 25 July 2020].

[40] Q. C., Y. L., S. H., and G. L., "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5105–5114.

[41] C. Qi, L. Yi, H. Su, and L. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017.

[42] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] P. K. Diederik and B. Jimmy, "Adam: A Method for Stochastic Optimization," in *In ICLR*, 2015.

[44] C. Yoo, S. Kim, S. Ji, Y. Shin, and S. Ko, "Capturing Hand Articulations using Recurrent Neural Network for 3D Hand Pose Estimation," *CoRR*, vol. abs/1911.07424, 2019. [Online]. Available: http://arxiv.org/abs/1911.07424

Van-Hung Le received M.Sc. degree at Faculty Information Technology Hanoi National University of Education (2013). He received PhD degree at International Research Institute MICA HUSTCNRS/UMI - 2954 - INP Grenoble (2018). Currently, he is a lecture of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; machine leaning, deep learning.
Email: van-hung.le@mica.edu.vn

Hai-Yen Tran received Bachelor degree at Faculty Information Technology National Economics University (2009). She received M.Sc. degree at Faculty Information Technology Hanoi National University of Education (2013). Currently, she is a lecture of Vietnam Academy of Dance. Her research interests include computer science, deep learning.
Email: yenth@vnad.edu.vn

INFORMATION FOR AUTHORS

Journal of Research and Development on Information and Communication Technology - *The MIC Journal of ICT Research*, or *ICT Research* in short – is a scientific publication of the Journal of Information and Communication, published by Vietnam Ministry of Information and Communications (MIC) since 1999. ICT Research is peer-reviewed, open-access, and published four times a year: two issues in *English* (ISSN: 1859-3534), and two issues in *Vietnamese* (ISSN: 1859-3526) (Chuyên san Các công trình nghiên cứu phát triển Công nghệ Thông tin và Truyền thông).

The purpose of *ICT Research* is to provide a forum for researchers and professionals to disseminate original and innovative ideas in the fields of information technology, communications, and electronics in Vietnam and worldwide. Its Editorial Board is composed of renowned scientists from research institutions throughout Vietnam and other countries.

AUTHOR GUIDELINES

Authors are encouraged to follow good practice of technical paper writing, such as the guidelines on "How to write for technical periodicals & conferences" of the Institute of Electrical and Electronics Engineers.

Submissions must be made online on the website of *ICT Research* [https://ictmag.vn]. Authors should format the manuscript as closely as possible to the *ICT Research* manuscript template.

By submitting a manuscript, authors are required to ensure that the submission has not been previously published, nor is currently submitted for publication elsewhere. Submission also implies that the corresponding author has the consent of all authors. The submission file must be in PDF format. To faciliate double-blind review, authors are requested to submit two versions of the manuscript, one with full information about all authors, the other without.

Submissions of revised manuscripts should follow the same guidelines as for the initial manuscript, plus with a document entitled "Response to reviewers' comments" which explains the modifications according to the comments of the anonymous reviewers.

Upon formal acceptance of a manuscript for publication, the authors are required to prepare the final manuscript in LaTeX, using the IEEE Transactions style.

PUBLICATION ETHICS

ICT Research is committed to following the rules, standards, and guidelines set forth by the Committee on Publication Ethics (COPE) in ensuring publication integrity and in handling cases of misconduct. All involved parties (Editors, Authors, Reviewers, and Publisher) are expected to be aware of and to conform to the respective rules, standards, and guidelines, available on COPE website.

The Editorial Board of *ICT Research* will immediately screen all articles submitted for publication. All submissions we receive are checked for plagiarism by using available online tools. Any type of plagiarism is unacceptable and is considered unethical publishing behavior. Such manuscripts will be rejected.

PEER REVIEW PROCESS AND PUBLICATION

An area editor of *ICT Research* (to be considered as the Editor from the point of view of the Authors) is assigned, by the Technical Editor-in-Chief, to each submission to coordinate the review process. The Editor pre-screens the submission and determines whether it is appropriate to the interests of readers of *ICT Research*. If appropriate, the Editor then initiates the review process. Each submission is then reviewed by at least two Reviewers. The review process is double blind, that is, identities of authors and reviewers are not revealed to each other.

Review decisions are: *(A) Accept submission* (i.e., no changes required), *(B1) Revision required* (i.e., minor revision), *(B2) Resubmit for review* (i.e., major revision, the submission needs to be re-worked with major changes and is then subject to a second round of review), and *(C) Decline submission* (i.e., reject). Each major round of review may take approximately 1 month. Should authors be requested by the Editor to revise the manuscript, the revised version should be submitted within 4 weeks for a major revision or 2 weeks for a minor revision. No more than 1 major revision and 2 minor revisions are allowed.

ICT Research is published with two issues a year in its *English Edition* (ISSN: 1859-3534). An accepted article will be immediately published online, with a DOI number, having been copyedited and laid out in compliance with *ICT Research* editing rules.

COPYRIGHT NOTICE

BÀI BÁO XUẤT SẮC SẼ ĐƯỢC CHỌN ĐĂNG TRONG CHUYÊN SAN SỐ ĐẶC BIỆT

THÔNG BÁO THAM DỰ HỘI THẢO QUỐC GIA

CITA 10

Từ 12-13/6/2021 tại Thành phố Đà Nẵng, Việt Nam

CITA (Conference on Information Technology and its Applications) là hội thảo khoa học quốc gia thường niên (bắt đầu từ năm 2012) về công nghệ thông tin và ứng dụng trong các lĩnh vực

Các bài báo xuất sắc nhất sẽ được chọn đăng trên số đặc biệt của Chuyên san "Các công trình nghiên cứu, phát triển và ứng dụng CNTT và truyền thông" (ISSN 1859-3526) của Tạp chí Thông tin và Truyền thông, Bộ Thông tin và Truyền thông.

Ban tổ chức

Trưởng Ban tổ chức
+ PGS.TS Huỳnh Công Pháp

Thường trực Ban chương trình
+ GS.TS Nguyễn Thanh Thủy
+ GS.TS Trương Bá Thanh
+ PGS.TS Lê Mạnh Thạnh
+ PGS.TS. Nguyễn Thanh Bình

Thường trực Ban tổ chức
+ TS. Trần Thế Sơn
+ TS. Nguyễn Quang Vũ
+ TS. Huỳnh Ngọc Thọ
+ TS. Phạm Nguyễn Minh Nhựt

Phụ trách xuất bản
+ TS. Nguyễn Quang Vũ
+ TS. Nguyễn Quang Hưng

Phụ trách tài chính
+ ThS. Nguyễn Thị Kim Ngọc
+ TS. Lê Thị Minh Đức

Bảo trợ chuyên môn

Ban cố vấn quốc tế

+ GS.TSKH Nguyễn Ngọc Thành (IEEE SMC Technical Committee on Computational Collective Intelligence; Wroclaw University of Science and Technology, Poland);
+ GS.TS Dosam Hwang (Department of Computer Engineering, Yeungnam University, Republic of Korea)
+ PGS.TS Lê Minh Hòa (Northumbria University, Newcastle, UK)

Các chủ đề

1. Khoa học dữ liệu và Trí tuệ nhân tạo
Chủ trì: PGS.TS Lê Anh Phương - TS. Lê Xuân Việt – TS. Phạm Xuân Hậu
2. Xử lý ảnh và ngôn ngữ tự nhiên
Chủ trì: PGS.TS Võ Trung Hùng – PGS.TS Nguyễn Tấn Khôi – TS. Phạm Thị Thu Thúy
3. Công nghệ phần mềm và Hệ thống thông tin
Chủ trì: PGS.TS Hoàng Quang – TS. Lê Thị Mỹ Hạnh – TS. Huỳnh Ngọc Thọ
4. Kinh tế số
Chủ trì: PGS.TS Đoàn Ngọc Phi Anh – TS. Lê Thị Minh Đức - TS. Nguyễn Thị Kiều Trang
5. Mạng và truyền thông
Chủ trì: PGS.TS Võ Nguyễn Quốc Bảo – PGS.TS Nguyễn Lê Hùng – TS. Trần Thế Sơn
6.. Chuyển đổi số và Đô thị thông minh
Chủ trì: TS. Hoàng Bảo Hùng – TS. Lê Thanh Hiếu - TS. Trần Thiện Vũ

Ngày hội dành cho Nghiên cứu sinh
Chủ trì: PGS.TS Nguyễn Gia Như – TS. Phạm Minh Tuấn – TS. Đặng Thanh Chương

Bài báo gửi đăng phải là bài viết nguyên gốc, chưa từng gửi đăng/xuất bản trước đó; MIỄN PHÍ hoàn toàn.

Đơn vị tổ chức

Chủ trì: VKU

Tham gia tổ chức HueCIT

CÁC MỐC THỜI GIAN QUAN TRỌNG:

- Hạn cuối nộp bài: 12/4/2021
- Thông báo chấp nhận: 12/5/2021
- Hội thảo: 12/6/2021

Website: http://vku.udn.vn/cita2021