A Repository of Method Fragments for Agent-Oriented Development of Learning-Based Edge Computing Systems

Iván García-Magariño, Moustafa M. Nasralla, and Jaime Lloret

Abstract

The upcoming avenue of IoT, with its massive generated data, makes it really hard to train centralized systems with machine learning in real time. This problem can be addressed with learning-based edge computing systems where the learning is performed in a distributed way on the nodes. In particular, this work focuses on developing multi-agent systems for implementing learning-based edge computing systems. The diversity of methodologies in agent-oriented software engineering reflects the complexity of developing multi-agent systems. The division of the development processes into method fragments facilitates the application of agent-oriented methodologies and their study. In this line of research, this work proposes a database for implementing a repository of method fragments considering the development of learning-based edge computing systems and the information recommended by the FIPA technical committee. This repository makes method fragments available from different methodologies, and computerizes certain metrics and queries over the existing method fragments. This work compares the performance of several combinations of dimensionality reduction methods and machine learning techniques (i.e., support vector regression, *k*-nearest neighbors, and multi-layer perceptron neural networks) in a simulator of a learning-based edge computing system for estimating profits and customers.

Introduction

Centralized machine learning (ML) approaches are difficult to apply in some Internet of Things (IoT) scenarios, given the rapid generation of data. In this context, learning-based edge computing systems (LBECSs) emerge as a solution for this context, allowing a distributed manner of performing ML.

However, the programming and coordination of the edge in LBECSs is not trivial and requires that programmers apply tailored solutions for applying and coordinating the different ML methods. This work proposes the adaptation of existing methodologies for multi-agent systems (MASs) with the existing ML methods for tailoring processes for developing LBECSs as MASs with ML on their nodes. The agent-oriented approach allows developing LBECSs with autonomous edge devices that can extract relevant information through ML techniques and share the relevant information concerning a required service. The agent-oriented approach also supports distributed coordination based on different flexible configurations of IoT devices with different types of sensors and tolerable to failures of some devices [1].

Several methodologies are proposed for the creation of MASs, such as Adelfe, Passi, and Ingenias. Their processes usually involve several steps, which can be described with several method fragments to facilitate the application of the processes and their further analysis.

The method fragments discussed in [2] allow designers to build their own processes by concatenating them. Thus, method fragments are the building units. As indicated in [3], method fragments need to be available for MAS designers in repositories. In this manner, designers can access them and build their processes.

The current work extends the repository of fragments [3] regarding the information proposed by the Design Process Documentation and Fragmentation working group of the IEEE Foundation for Intelligent Physical Agents (FIPA) Methodology Technical Committee (TC).

Both the previous repository and the proposed extension are aimed at allowing designers to acquire quick and easy retrieval of fragments, but they use different definitions for fragments. The previous repository is based on the old definition of fragment (the FIPA one), and presents an easy and simple interface providing means only for selecting fragments on the basis of their categorization, that is, activity, process role, work product, and kind of MAS meta-modeling entity (MMME). However, in the new application, designers can use the new definition of fragments that is indicated in [2] and is called *method fragment*.

Furthermore, this new repository extension computerizes the measurement of fragments' *attitude* to specific features (e.g., autonomy and sociability). The attitude reflects the degree to which a fragment can imply that the work products (i.e., MASs) have a given feature.

Finally, the repository is tested by inserting not only PASSI method fragments but also INGENIAS method fragments. More concretely, as experimentation, we have developed a simulator of an LBECS called BigDataSim in order to illustrate the current approach. In this experimentation,

Digital Object Identifier:
10.1109/MNET.011.2000296

Iván García-Magariño is with Complutense University of Madrid, and also with the Instituto de Tecnología del Conocimiento; Moustafa M. Nasralla is with Prince Sultan University; Jaime Lloret is with the Universitat Politècnica de València.

we have compared the different dimensionality reduction methods and several ML techniques, and we have compared the errors.

The remainder of the article is organized as follows. The next section discusses the related work. We then introduce the background of this work. Following that, we describe the repository database for method fragments in the development of MASs. Then we present the experimentation with the LBECS simulator and comparison of ML methods. Finally, we discuss the conclusions and future work.

Related Work

A recent work proposed an approach of model-driven development of agent-based cyber-physical systems [4]. Their approach used the Organization-Based Multi-Agent Software Engineering (O-MaSE) process framework using method frameworks. However, their approach was mainly focused on cloud computing rather than edge computing.

In addition, there is a repository for method fragments that are represented with the Open Process Framework (OPF). Furthermore, the work in [5] presented a Situational Method Engineering (SME) framework that contains a repository for reusing situational method fragments. That framework allows designers to combine fragments, and computerizes the retrieval of method fragments that have certain requirements.

Nevertheless, none of those works have computerized the measurement of metrics for further analysis of processes that contain certain process fragments as the current work has done.

Anya *et al*. [6] presented an agent-based simulator (ABS) that simulates the discovery of big data. Their simulations showed that cross-group knowledge exchange increases the discovery from data in comparison to collaboration only within each group. They provided an initial ABS approach for simulating the influence of organizational dynamics and social structures on the discovery and reuse of relevant data.

Some researchers on simulation are now focusing on obtaining high-performance frameworks for modeling and simulations for supporting big data applications [7]. In this line of research, the Care HPS (i.e., from high-performance simulation) tool [8] supports agent-based modeling with parallel computing for increasing the performance of the simulations.

Big data have assisted enterprises in increasing their profits. For instance, the work in [9] proposed the metric profit per hour as an indicator for supporting decision making in business. Their approach was able to measure this metric thanks to big data analytics. Their approach was designed to be applicable to all manufacturing industries.

Big data has also been useful for discovering lifetime value of costumers. Chiang et al. [10] applied big data analytics to determine the relations of personality traits and country of origin of customers with their potential as lifetime customers. They found that customers with peacefulness and openness personality traits were the ones with the highest customer lifetime values. Thus, this information can be used for determining useful market strategies.

Big data can be applied to customize services for customers. For example, the approach of [11] analyzed customer behaviors by means of big data analytics. They proposed an implementation based on the Map Reduce model with the Hadoop technologies to analyze the data, and visualized the results through an HTML interface.

In the field of wearable technology, Chen *et al*. [12] proposed an LBECS that coordinated different wearable devices. Each wearable device applied deep analytics for performing ML on the edge and sharing their knowledge discovery with other devices in order to propose complex services to the user. This work focused on the challenges related to the integration of different types of edge wearable devices and its underlying architecture, but did not provide support for composing development processes.

Some of the aforementioned works proposed generic method engineering approaches, others present different ML contributions, and others focus on integration challenges in LBECSs. Nevertheless, none of the aforementioned works proposed to use method engineering for improving the development processes of specifically LBECSs.

Background in Learning-Based Edge Computing and Method Engineering

The LBECSs are at the intersection of ML and edge computing. Some works have applied tailored solutions of applying ML in edge computing. In particular, Kozik *et al.* [13] presented a scalable ML approach for detecting attacks in edge computing. They proposed an LBECS as they performed the ML on the nodes to detect upcoming attacks. More concretely, they used a pre-built classification model, but they did not have enough processing and storage capabilities. Thus, they applied traffic classification using extreme learning machines for coordinating the ML across the different edge nodes. Nevertheless, they did not propose any software engineering methodology for developing LBECSs as the current work does.

Method engineering was defined as the engineering discipline to design, construct, and adapt methods, techniques, and tools for the development of information systems. Method engineering is related to: meta-modeling techniques that describe modeling languages; tool interoperability, in which fragments are recommended to be available in repositories and to be combined; situational methods, for describing and particularizing the methods to specific development projects; and comparative review of methods and tools.

A *method fragment* is a discrete component of a method that can be used in one or more methods. The discreteness of method fragments makes it much easier to describe their strengths and weaknesses than is the case with methods, enabling designers to put together a customized method to suit a particular elicitation need.

The agent-oriented software engineering (AOSE) methodologies have been divided into method fragments for their analysis, promotion, and application. FIPA describes the transition from general method engineering to its specific application in AOSE methodologies. That work is promoted by the FIPA Methodology TC, and indicates

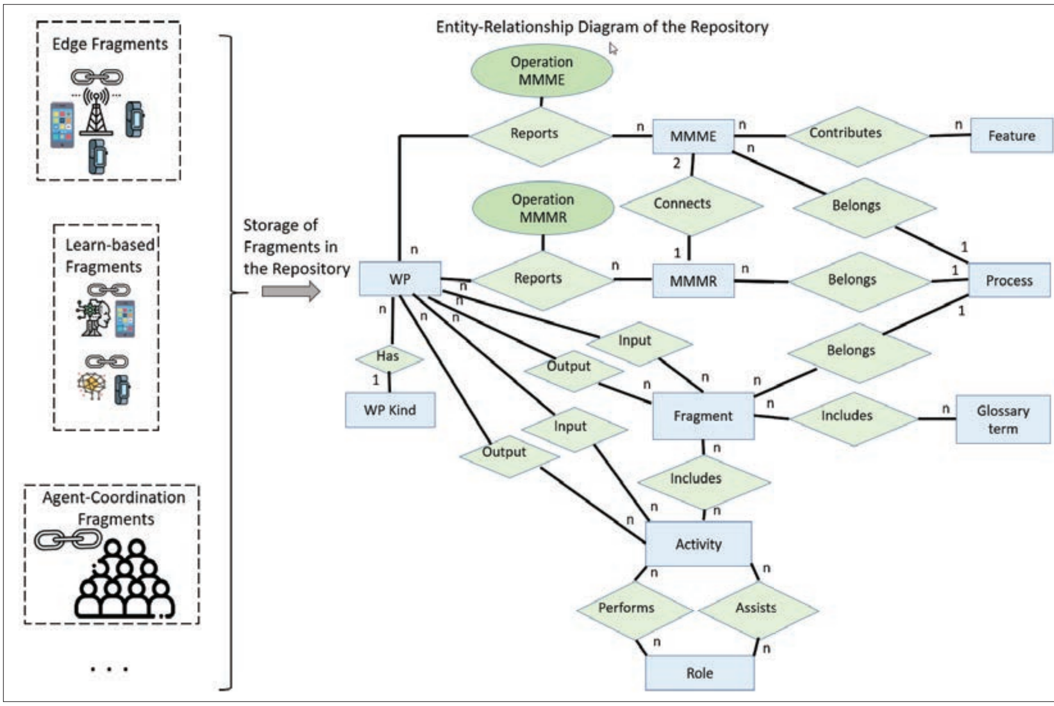

FIGURE 1. The entity-relationship design of the repository database for LBECSs.

the structure of documents for describing method fragments for AOSE methodologies. This document structure includes the definition of input and output work products (WPs) and the concepts to be designed in each fragment. Among other features, the structure also includes the relationships between the WPs and the MAS meta-modeling elements (MMMEs), and a composition guideline for each fragment. The information of that structure is included in the presented repository.

Furthermore, it is worth mentioning that the method fragments of several AOSE methodologies are already described with the document structure promoted by the FIPA Methodology TC and presented in [2]. Specifically, an example of a PASSI fragment is included in [2] and the description of PASSI fragmentation as another example. The ADELFE method fragments and INGENIAS are already defined.

The Repository for Learning-Based Edge Computing Systems

This repository stores method fragments for AOSE and all the data involved with the method fragments. This repository is structured by taking the recommendations of the FIPA Methodology TC [2] into account, and the design of this repository is further described below.

The repository is implemented by means of a database. This database is complemented with menus and formularies in order to facilitate the human interaction with the database, and these menus and formularies are presented below.

The analysis of AOSE methodologies can become crucial when selecting fragments for composing a development process. For this reason, this repository includes queries that computerize the measurement of certain metrics that facilitate the association of processes with attitudes, such as the autonomy and the sociability. We briefly introduce these metrics and the corresponding formularies that computerize their measurement below.

Design

The entity-relationship design of the repository database is presented in Fig. 1. The repository includes all the information recommended by the FIPA Methodology TC: fragments, processes, input and output WPs of fragments, WP kinds, activities, roles, and glossary terms. Each WP is linked with MMMEs and MAS meta-modeling relationships (MMMRs), with an operation mode that can be designed, related, and quoted.

Furthermore, MMMEs are related to certain features (e.g., autonomy and sociability) with certain weights in order to analyze the features of processes, as described later.

Each agent using ML in each edge device has a purpose in providing each service. Depending on the service that is being provided, each agent plays a role for providing this service. This role is usually performed by several agent activities and assisted by other activities.

The Running Repository System

The repository system incorporates menus and formularies for facilitating its use. The database contains menus that organize all the existing operations. At the main menu, the *Process* and *Fragment* options are concerned with the storage and management of method fragments, whereas the *Feature* and *Metrics* options are regarding the analysis and measurement of the existing method fragments.

Figure 2 presents an excerpt of the form for fragments. As one can observe, the form contains the information proposed by the FIPA Methodology TC. There is an introduction, in which the user can both write a text and link an external document containing a figure of the whole process to which the fragment belongs. The form also contains the description, consisting of a text and a reference to an external document. In the form, users can also specify the workflow, a guideline, the composition guideline, the aspects, the dependency relationships, and the process to which the fragment belongs.

Within the form for fragments, a user can also include a picture of the MAS metamodel. Furthermore, the form of fragments includes a sub-form for managing its activities. This sub-form includes a button for opening the activity form in case the user needs to create a new activity.

The form of fragments also contains a sub-form for WPs, indicating the input and output WPs. The form also includes buttons for accessing the formularies that relate the WPs with their MMMEs and MMMRs. In this manner, designers can change and observe the WP that they are adding to a given fragment.

Moreover, the form for fragments also includes a sub-form for managing its glossary terms, as recommended by the FIPA Methodology TC.

As a proof of concept, fragments are introduced in the repository for the INGENIAS [14] and PASSI [2] methodologies.

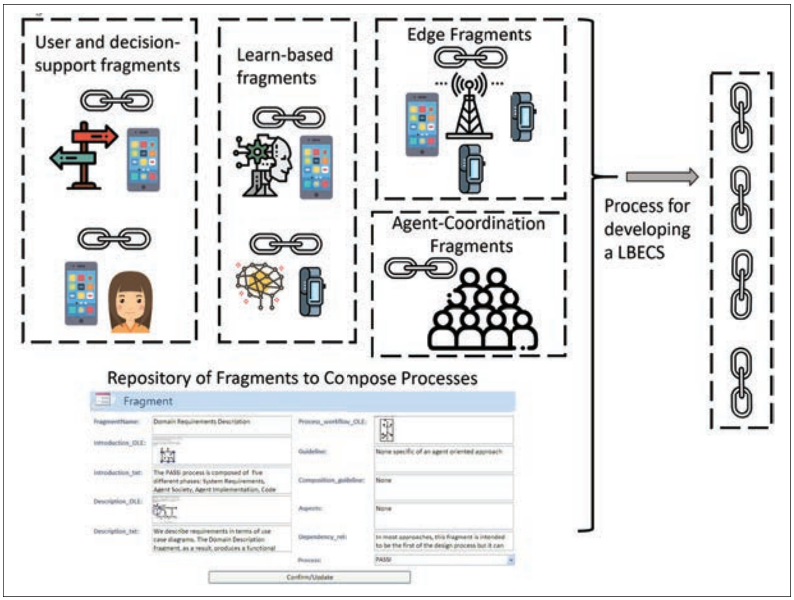

FIGURE 2. Definition of fragments in the repository for composing processes to develop LBECSs.

Measurement and Analysis

A hypothesis of another of our ongoing works is that the importance of an element can be measured with the element relevance index (ERI), which is calculated in Eq. 1.

$$ERI(E) = NumRel(E) + NumFrag(E) + NumWP(E) \quad (1)$$

where:

- *NumRel(E) is the number of MMMRs in which the E element is involved.*
- *NumFrag(F) is the number of fragments that are related to the E element.*
- *NumWP(F) is the number of work products that are related to the E element.*

ERI is relevant for calculating the features of processes composed of certain method fragments, such as the sociability of agents from different devices, showing the kind of expected communication. For instance, processes with intensive communications will probably need an appropriate communication infrastructure, while systems with computing-intensive ML techniques like deep learning will probably need devices with enough computation resources.

The query *MeasureERI* computerizes the measurement of ERI, in which the user just selects the MMME and the query returns its ERI. This query and other queries that facilitate obtaining partial results are summarized below:

- MeasureERI: It measures the ERI of a given MMME according to Eq. 1.
- ERI: It provides a list of all the names of all the MMMRs, fragments, and WPs related to a given MMME.
- NumRel: It provides a list of all the names of all the MMMRs related to a given MMME.
- NumFrag: It provides a list of all the names of all the fragments related to a given MMME.
- NumWP: It provides a list of all the names of all the WPs related to a given MMME.

The ERI measurement can be calculated with the presented repository by means of the form shown in Fig. 3. This figure presents a brief practical example of the measurement of ERI over the InteractionUnit: entity. This entity represents a message among two different devices. Its relevance (i.e., ERI = 5) shows the need of a proper communication infrastructure for the process designed for the development of health LBECSs in smart hospitals.

The Methodology Attitude (MA) metric calculates the degree in which a given methodology is related to a given feature, such as autonomy or sociability. For this reason, the MMMEs are related with attitudes indicating weights. Then the MA metric is calculated for a given process with a weighted sum of the ERI measurement values of the related MMMEs. Figure 4 shows the form that calculates the MA metric.

Experimentation with Learning-Based Edge Computing in the BigDataSim Simulator

The objective of this use case was to use the repository to compose a process with method fragments for a tailored development of an LBECS in the field of business boosted by analysis of big data from IoT sensors. Functional requirements referred to the use of IoT sensors information in shops to extract relevant data that can help in shop organization to boost profits. Non-functional requirements limited the amount of data transmitted over the network so that most processing should be performed locally on the edge, and established a minimum percentage of 10 percent for the increase of profits.

This experimentation has developed an ABS of an LBECS called BigDataSim to illustrate the proposed approach, using the process created from the repository. BigDataSim was designed to fulfill both aforementioned functional and non-functional requirements. It simulated the sales of different shop organizations, incorporating shop organization improvements based on the learning from user behaviors detected with IoT sensors. It was trained with different techniques of

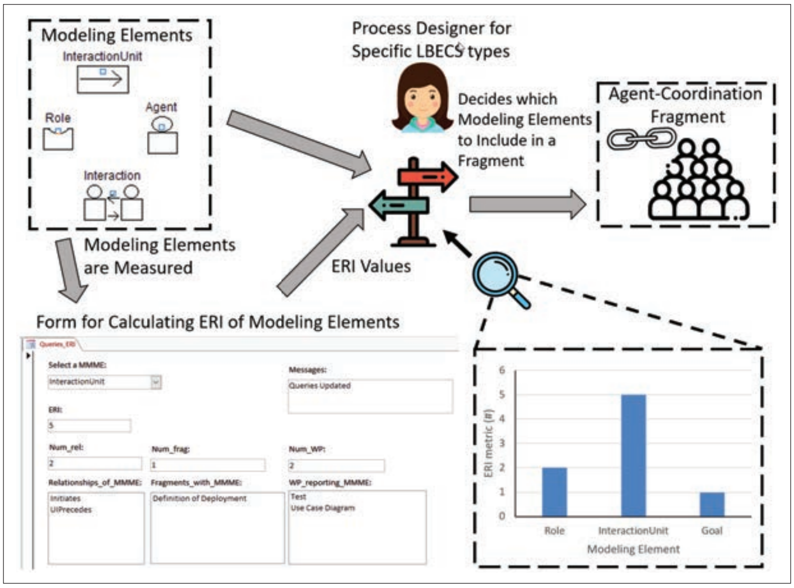

FIGURE 3. Calculation of ERI for selecting modeling elements for each method fragment for LBECSs.

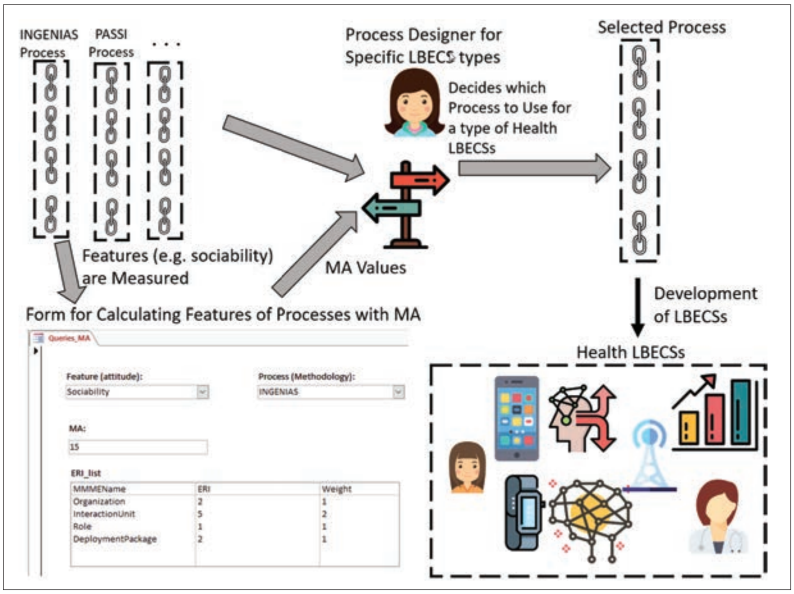

FIGURE 4. Calculation of MA for selecting a process for a type of health LBECS.

ML and dimensionality reduction on several simulated edge nodes, for estimating the number of customers and profits, based on IoT devices that track people entering and sales. For example, certain sensors can measure the number of customers that go through the doors of a shop to count the number of customers that visit a shop. The cash registers can track the number of customers actually buying items. In this way, the LBECS can calculate the ratio of customers that left the shop without buying anything. Presence sensors can also estimate in which section customers are looking for items to estimate which products customers may be missing and could be more profitable to include, with most of the data processing on the edge. Then the ML models were shared in the different nodes to improve their accuracy. This LBECS has been developed using the proposed method engineering approach for applying the combination of different well-known AOSE methodologies.

The presented ABS provides a mechanism to estimate the profitability of including big data analytics in a firm. The user introduces certain input parameters about the potential customers, possible repercussions of marketing strategies, estimated costs, and duration of the simulation. The ABS simulates different kinds of customers and the repercussions on the marketing campaigns of two different firms.

One firm (referred to as control firm hereafter) performs a marketing campaign toward some particular people from potential customers offering very special bargains. These potential customers are selected for a particular age and with certain features. The selection of the specific people from potential customers is random as the firm only has general information.

The other firm includes big data analytics and has a method to estimate the customer lifetime value (CLV). This value is an estimated probability of the customers becoming lifetime users. This firm is assumed to be able to follow particularized marketing strategies for gaining the same customers as the other user per month. The difference is that it selects the customers with high CLV. However, this company has a cost for including big data analytics.

The ABS shows the comparative results of the evolution of profits and customers. In this way, firm owners can have an estimation of which might be the most profitable option between whether or not to include big data analytics.

The main kinds of agents are "customer" agents and "firm" agents. In the case of firm agents, there is an extension for simulating firms with the use of big data analytics called "firm-big-data" agents. Firm agents were able to apply common ML methods such as support vector regression (SVR) and neural networks. Firm-big-data agents not only used ML methods but also incorporated the possibility of applying dimensionality reduction techniques. These facilitate reducing the information of big datasets for efficiently applying ML methods. We considered this separation for properly separating ML methods from big data techniques so that the former could be reused separately in the future.

The ABS is designed in a generic way for being applicable to most kinds of markets. However, the ABS simulates only one market type in each simulation. For this reason, customers are modeled to simulate their relation with only one kind of market.

Each customer agent can be associated with a firm or not. Each customer agent has a reference to the firm from which it is buying some service or goods, if any. Normally, the customer agent can register in a firm. However, the customer agent hires the service for a certain period. Then it can become a lifetime customer or leave the firm. This is simulated with an agent-based nondeterministic decision as proposed by the technique for developing agent-based simulation apps and online tools with nondeterministic decisions (TABSAOND) [15]. This decision is simulated by the generation of a random number and its comparison to certain probability. The CLV of the customer agent determines this probability.

A decision d = 1 determines that the agent becomes a lifetime consumer of its actual firm,

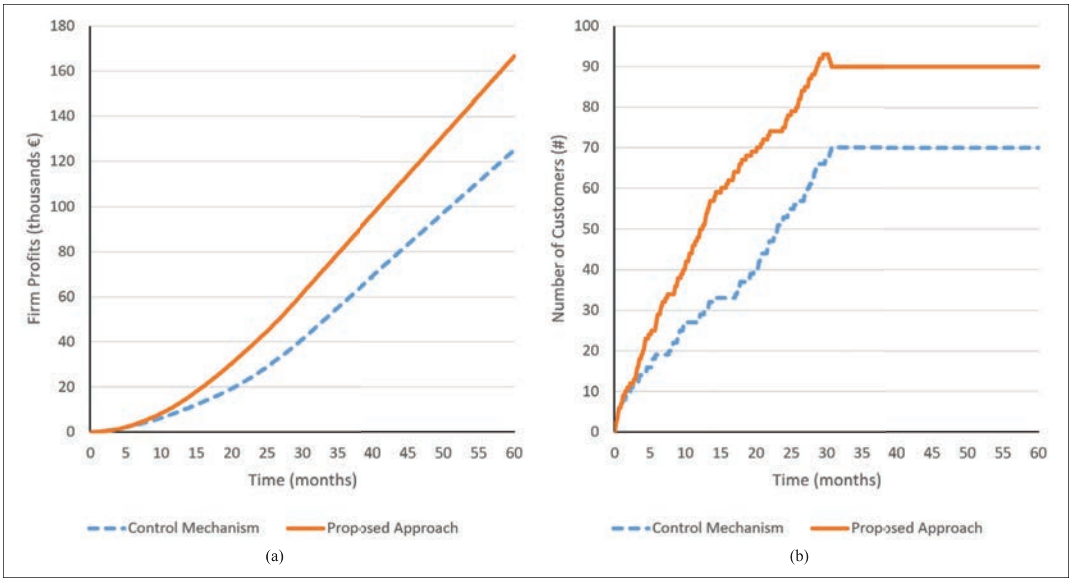

FIGURE 5. Charts of simulation evolutions in BigDataSim for a) profits; b) customers.

while $d = 0$ determines that the consumer unsubscribes from the firm agent once the common duration of a non-lifetime consumer expires.

If the customer does not become a lifetime user, it unsubscribes from the firm's service and becomes available again in the market. The customer can have a high CLV and a low CLV. The ABS user can determine the number of potential customers with low and high CLVs with some input parameters.

The firm agent simulates the management of the corresponding company. It manages the profits of the firm starting this variable as zero at the beginning of the simulation. Then it adds the profits per month proportionally distributed in the iterations of the month, considering the multiplication of the number of current customers and the profits per customer. Thus, the profits earned in each iteration (referred to as p_i) are calculated with the corresponding ratio, with a given number of customers, the profits earned from each customer per month, and the number of iterations per month.

The calculation of the accumulated profits (p) is performed by summing the profits per iteration. The firm big data agents also calculate their profits per iteration in the same way. However, these also calculate the costs of big data per iteration (denoted as c_i). The update of the profits per month in firm big data agents also takes these big data costs into account, with the sum of all these values.

Users can interact with BigDataSim through its user interface (UI). A user can introduce the input parameters. The user indicates an estimated amount of the number of potential customers with a low CLV and an estimated number of the ones with a high CLV. Regarding the market and the previous experience, the user introduces an estimated amount of the number of customers that the firm can gain per month using particularized marketing strategies. The user also determines the estimated profits from each customer per month.

The user is normally thinking about hiring a person for performing the big data analytics or hiring the service of a third party. Then the user must introduce the cost of applying big data per month. In addition, the user indicates the common duration of non-lifetime customers regarding the simulated market. Finally, the user introduces the time of the simulation expressed as the number of months. The user can press the "Run simulation" button to start the simulation.

The UI of BigDataSim shows the simulation evolution. In particular, there are two bottom buttons that allow users to switch between two different views. The first view shows the evolution of profits in thousands of euros in a chart like the one shown in Fig. 5a. It shows the evolutions of the firm without using big data analytics and the firm when using this analytics, respectively. In this way, the user can compare the estimated profits depending on whether or not big data analytics are used. Figure 5b shows the second view of the evolutions of the numbers of customers in another chart. This chart also simulates the two possibilities of whether or not to use big data analytics.

In order to fully experience this approach, we have compared the average errors of the different combinations of dimensionality reduction methods and ML methods. Among the existing dimensionality reduction methods, we used recursive feature elimination (RFE), non-negative matrix factorization (NMF), and feature selection (FS) with a variance threshold. Among existing ML techniques, we applied SVR, *k*-nearest neighbors (kNN), and a multi-layer perceptron (MLP) neural network. We also used linear regression as a basis comparison. Figure 6 shows the boxplots of these errors. In this way, readers can check the average and different quartiles, as well as observe the outliers. Notice that the combination of SVR and FS obtained the lowest average errors among the combinations of ML methods and dimensionality reduction techniques. This combination improves the performance of the edge learning process, almost without increasing the error over applying SVR without dimensionality reduction, so this combination is appropriate for developing efficient and accurate LBECSs in this context. In contrast, the highest increase of error was obtained when applying MLPs with NMF, so this combination is not recommended in this context.

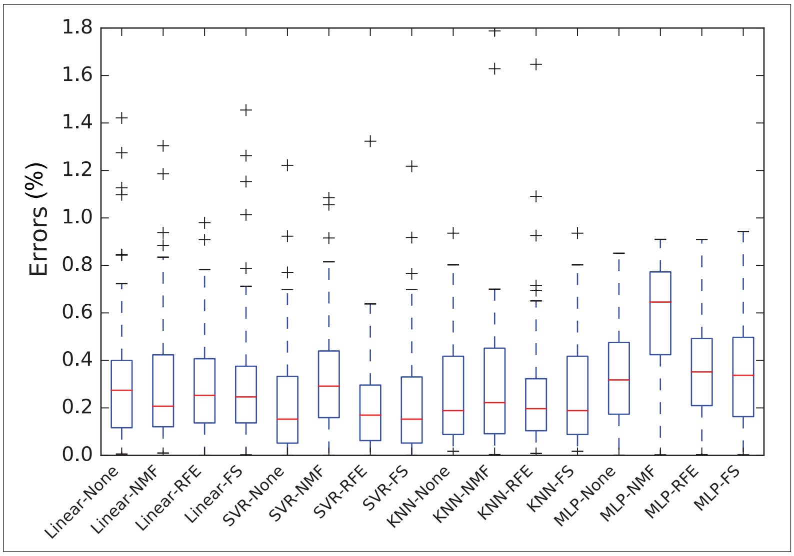

FIGURE 6. Boxplot of errors of the different dimensionality reduction and ML methods in edge computing.

Conclusions and Future Work

This article has presented a new repository for AOSE method fragments in order to build LBECSs including ML methods in the development of edge computing software. Their main achievements are first to technically support the storage and share of the fragments and second to computerize the measurement of metrics for analyzing the features of processes composed of some given fragments.

To illustrate this approach, we have developed the BigDataSim simulator for testing different ML configurations in edge computing applications. This simulator has been applied in the context of estimation of profits and number of customers in supermarkets based on the received data from IoT sensors. We simulated this LBECS with different combinations of ML methods and dimensionality reductions, and provide a comparison of their resulting errors.

As future work, we plan to apply the proposed approach for developing an LBECS for managing smart cupboards for measuring the memory of people, identifying their emotions, and providing dietary suggestions in an intelligent and supervised way. Thus, the smart cupboards will learn from their common users with the proposed approach. We plan to compare the proposed approach with alternative methodologies by comparing their efficacy in developing the smart cupboard system to further assess the improvement of the proposed approach over the alternatives.

Acknowledgments

The authors acknowledge PSU Smart Systems Engineering Lab, project "Utilisation of IoT and sensors in smart cities for improving quality of life of impaired people" (ref. 52-2020), CYTED (ref. 518RT0558), and the Spanish Council of Science, Innovation and Universities (TIN2017-88327-R).

References

[1] G. Fortino *et al.*, "Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, 2017, pp. 1939–56.

[2] M. Cossentino *et al.*, "Method Fragments for Agent Design Methodologies: From Standardisation to Research," *Int'l. J. Agent-Oriented Software Engineering*, vol. 1, no. 1, 2007, pp. 91–121.

[3] V. Seidita, M. Cossentino, and S. Gaglio, "A Repository of Fragments for Agent Systems Design," *Proc. Wksp. Objects and Agents*, 2006, Catania, Italy, pp. 1–8.

[4] I. Batchkova and T. Ivanova, "Model-Driven Development of Agent-Based Cyber-Physical Systems," *IFAC-PapersOnLine*, vol. 52, no. 25, 2019, pp. 258–63.

[5] I. Mirbel and J. Ralyté, "Situational Method Engineering: Combining Assembly-Based and Roadmap-Driven Approaches," *Requirements Engineering*, vol. 11, no. 1, 2006, pp. 58–78.

[6] O. Anya *et al.*, "Understanding the Practice of Discovery in Enterprise Big Data Science: An Agent-Based Approach," *Procedia Manufacturing*, vol. 3, 2015, pp. 882–89.

[7] J. Kolodziej, H. González-Vélez, and H. D. Karatza, "High-Performance Modelling and Simulation for Big Data Applications," 2017, pp. 1–2.

[8] F. Borges *et al.*, "Care HPS: A High Performance Simulation Tool for Parallel and Distributed Agent-Based Modeling," *Future Generation Computer Systems*, vol. 68, 2017, pp. 59–73.

[9] M. Hammer *et al.*, "Profit Per Hour as a Target Process Control Parameter for Manufacturing Systems Enabled by Big Data Analytics and Industry 4.0 Infrastructure," *Procedia CIRP*, vol. 63, 2017, pp. 715–20.

[10] L. L. Chiang and C. S. Yang, "Does Country-of-Origin Brand Personality Generate Retail Customer Lifetime Value? A Big Data Analytics Approach," *Technological Forecasting and Social Change*, 2016; https://doi.org/10.1016/j.techfore.2017.06.034.

[11] A. A. Khade, "Performing Customer Behavior Analysis Using Big Data Analytics," *Procedia Computer Science*, vol. 79, 2016, pp. 986–92.

[12] M. Chen *et al.*, "Living with I-Fabric: Smart Living Powered by Intelligent Fabric and Deep Analytics," *IEEE Network*, 2020.

[13] R. Kozik *et al.*, "A Scalable Distributed Machine Learning Approach for Attack Detection in Edge Computing Environments," *J. Parallel and Distributed Computing*, vol. 119, 2018, pp. 18–26.

[14] J. Pavón and J. Gómez-Sanz, "Agent Oriented Software Engineering with INGENIAS," *Multi-Agent Systems and Applications III*, ser. LNCS, vol. 2691, Springer, 2003, pp. 394–403.

[15] I. García-Magariño *et al.*, "TABSAOND: A Technique for Developing Agent-Based Simulation App. And Online Tools With Nondeterministic Decisions," *Simulation Modelling Practice and Theory*, vol. 77, 2017, pp. 84–107.

Biographies

Iván García-Magariño is a lecturer at the Complutense University of Madrid. Prior to this job position, he was a Ph.D. assistant professor at the University of Zaragoza (2014-2018), and before that he was a lecturer at the Madrid Open University (2010-2014). He was awarded his Ph.D. in computer science engineering in 2009, and was a recipient of an FPI scholarship from 2006 to 2010. His main research interests include AI, human-centric AI, agent-based simulators, multi-agent systems, and cryptocurrency/blockchain. Among journals, book chapters, conferences, and workshops, he has over 140 publications (over 65 in journals with ISI Thomson JCR).

Moustafa M. Nasralla received his B.Sc. degree in electrical engineering from Hashemite University, Jordan, in 2010, his M.Sc. degree (Hons.) in networking and datacommunications from Kingston University London, United Kingdom, in 2011, and his Ph.D. degree from the Faculty of Science, Engineering and Computing, Kingston University. He is currently an assistant professor at Prince Sultan University. His research interests include the latest generation of wireless communication systems, for example, 5G, LTE-A, LTE wireless networks, M2M, the Internet of Things, machine learning, OFDMA, and multimedia communications.

Jaime Lloret [M'07, SM'10] received his M.Sc. degree in physics in 1997, his M.Sc. degree in electronic engineering in 2003, and his Ph.D. degree in telecommunication engineering (Dr.Ing.) in 2006. He is currently an associate professor with the Polytechnic University of Valencia. He is the Chair of the Integrated Management Coastal Research Institute and the head of the Active and Collaborative Techniques and Use of Technologic Resources in the Education (EITACURTE) Innovation Group. He was the Internet Technical Committee Chair (IEEE Communications Society and Internet Society) for the term 2013–2015. He has over 480 research papers (over 230 with JCR).