

# An Architectural Schema for Performance Prediction using Machine Learning in the Fog-to-Cloud Paradigm

Souvik Sengupta  
CRAAX Lab, DAC  
UPC BarcelonaTech  
Vilanova i la Geltrú, Spain  
souvik@ac.upc.edu

Jordi Garcia  
CRAAX Lab, DAC  
UPC BarcelonaTech  
Vilanova i la Geltrú, Spain  
jordig@ac.upc.edu

Xavi Masip-Bruin  
CRAAX Lab, DAC  
UPC BarcelonaTech  
Vilanova i la Geltrú, Spain  
xmasip@ac.upc.edu

Andrés Prieto-González  
CRAAX Lab, DAC  
UPC BarcelonaTech  
Vilanova i la Geltrú, Spain  
aprieto@ac.upc.edu

**Abstract**—The Fog-to-Cloud (F2C) paradigm is emerging to both provide higher functional efficiency for latency-sensitive services and also help modern computing systems to be more intelligent. As it is still in its infancy, the biggest challenge for this domain is to build a proper resource allocation technique as part of an efficient resource management module. The diversified and distributed nature of that paradigm creates some additional hurdles for choosing the appropriate resources for executing some tasks. Significantly, efficient resource consumption estimation and performance forecasting are core issues in the design and development of a proper and smart resource management mechanism for F2C systems. Considering this fact, in this paper, we aim at designing an architectural framework for a prediction-based resource management mechanism for F2C systems. The performance prediction is based on supervised machine learning technology. The proposal has been evaluated and validated by predicting the performance and resources usage of F2C resources through several tests. Primarily, we have run an image recognition application on different F2C resources and collected performance-related information and resource consumption information. Then, by adopting the multivariate regression methodology, we perform some standard machine learning techniques to predict the performance and estimate the resource consumption of the F2C resources. Finally, to justify the effectiveness of our proposal, we calculated the value of a cost function between estimated values and the real measured values.

**Index Terms**—Fog-to-Cloud (F2C), Internet-of-Things (IoT), performance prediction, performance forecasting, resource management.

## I. INTRODUCTION

Fog-to-Cloud computing (F2C) emerges to provide the facilities of both cloud and fog/edge computing among the end-users. Inheriting the strength of both computing paradigms, F2C ensures better and improved performance for latency-sensitive smart services [1]. Being the F2C still in a developing stage, authors in [1] argue that an exhaustive and intelligent

resource provisioning and management mechanism is strongly required. Indeed, for any distributed computing system, an efficient resource provisioning and management mechanism is a mandatory commodity [2] to make the whole system to become more productive and intelligent. Individually, either in the cloud (e.g., [3], [4]) and in the fog/edge (e.g., [5], [6]) computing platforms, a significant amount of research works have been done for designing appropriate and efficient resource provisioning and management mechanisms.

In any distributed computing platform, on-demand resource allocation is a critical process to design the proper and intelligent resource provisioning mechanism for that platform [7]. For any smart and distributed computing system, it is a challenging job to efficiently allocate available system resources based on their current and future demand for executing some computational tasks. For example, the authors in [8] classified the cloud-computing resource provisioning mechanism into two categories 'Predictive' and 'Reactive'. According to them, in case of the 'Reactive' approach first, the current state of system resources has to be measured before allocating the system resources. Then following the current task requests, available system resources are allocated. Whereas, in the case of 'Predictive' approach, based on the system's historical information and upcoming tasks information, the resources requirement has been calculated. Interestingly, in our literature, we found many works (e.g. [9]–[11]) where researchers are following the 'Predictive' approach to ensure better resource utilization for their smart and distributed computing platforms.

In the case of 'Predictive' approach, as the system is able to forecast the computational load in advance and also able to forecast the overall performance of system resources, this approach can help to design a better resource provisioning and management mechanism [6]. Interestingly, the 'Predictive' approach is one of the basic requirements, which helps

any computing system to be much smarter [12]. Since F2C is still in an emerging and developing stage, we have not found any research work, which has been done so far for predicting the resource usage and performance in combined F2C computing scenarios. To address this challenge and build an efficient resource management mechanism in the coordinated and combined F2C computing platform, in this paper we present an architectural framework for predicting the performance of F2C resources. To that end, we adopt machine learning for predicting the performance of F2C resources. Moreover, in order to ensure better resource utilization in the F2C computing paradigm, we use existing machine learning techniques to predict the usage of F2C resources for executing some tasks.

To this end, the rest of the paper is organized as follows. In Section II, by focusing on some existing work, we define the problem statement of our work. After that, in Section III, we briefly describe the architectural schema of the F2C paradigm. Then in Section IV, we present the proposed framework for predicting the F2C resource usage and performance. In Section V, by performing some tests, we evaluate the effectiveness of our proposal. Finally, we conclude our paper by presenting future directions and concluding remarks.

## II. STATE OF ART

The key objective for building a predictive approach based resource management mechanism is to accurately forecast the resource usage and performance in advance; so that, smartness level of the considering computing paradigm can be improved. Significantly, to fulfill this purpose, a vast number of research works have been already done in the cloud computing domain (e.g., [13]–[17]) aimed at improving the resource management strategy. Likewise, in other computing domains (i.e., smart grids, mobile ad-hoc networks) [18], [19], we also found some works, where researchers show their concerns to forecast the resource usage to reduce the workload for their system resources.

Interestingly, we also found some research works in the IoT and edge/fog computing domains (i.e., [20], [21]), where researchers put the focus on estimating the resource usage for upcoming tasks to reduce the load of their system resources. Even, we found that a significant amount of research works (e.g., [22]–[26]) have been done in various other computing platforms, where researchers proposed some ML techniques for improving the resource provisioning and management.

After reviewing these works, we found one common thing, mentioned by them all; that is data. Most importantly, without the data, it is impossible to train any machine learning model. Even the selection of wrong data can easily reduce the accuracy-level of the trained ML model. So, for that reason, it is necessary to process the data and clean it. Unfortunately, we have not found any work where researchers performed some operations or presented some techniques to either clean or process the system data (i.e., data pre-processing) for building and training their ML model.

Furthermore, for edge or fog computing the prediction has to be done near to the end-users. For that reason, the data have to reside near to the processing node; but unfortunately, we have not found any prior work to address this issue. Also, in most of the cases, researchers provide a more specific solution to improve their prediction model for their proposed systems. For example, in [13] researchers use an existing ML approach, to build some prediction model for improving the forecasting of resource usage in the cloud computing platform. In case of distributed, combined and hierarchical computing platform (i.e., F2C), it is necessary to define the architectural framework for predicting the resource usage. Sadly, in our literature study, we have not found any work where researchers focus on building or designing an architectural framework for prediction-based resource management in a hierarchical, distributed and combined computing platform. Therefore it is essential to define and design that kind of architectural framework for combined F2C systems, which is a yet demanding specific effort from the research community.

## III. ARCHITECTURAL DESCRIPTION

The benefits of a F2C-based computing platform has been already described in [1]. Also in our past work [27], we have already justified how the F2C-based execution model is becoming a predominant computing architecture to offer the better service facilities and experiences for any latency-sensitive applications. According to the authors [1], by ensuring a much better service execution and optimally utilize the system resources, F2C helps to bring more smartness for any computing system.

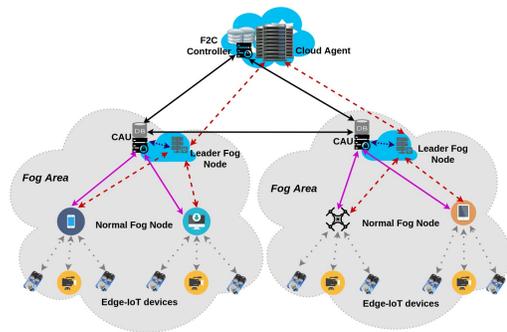


Fig. 1. Distributed DB-based F2C execution model

The F2C has been presented as a combined, layered and hierarchical computing platform [1], where the cloud resources are residing in the uppermost layer of the platform and the bottom-most layer contains full of sensor and small IoT devices. In between these two layers, the fog layer exists, putting together resources not so enriched like cloud resources. The fog layer is responsible for bringing the cloud facilities near to the edge of the network. By performing the computational tasks near to the end-users or at the edge of the network, the fog layer resources ensure better service facilities and experiences for latency-sensitive services and help to bring more smartness for any modern computing platform (e.g.,

smart city, industry 4.0). In our previous work [27], we already presented an improved version of the F2C computing architecture for any smart platform (e.g., smart city, industry 4.0). In that paper [27], we explained and justified how the secure data distribution over the fog layer improved the task execution time and offered a better service experience among the end-users of any smart platform (e.g., smart city).

In that paper we have already comprehensively described the architecture of the progressive F2C paradigm and also discussed the different layers and resource components, which are potentially involved into our proposed computing paradigm, for an specific smart city scenario. According to that work, in Fig. 1, we depict the architectural representation for the infrastructure of an F2C-based smart city scenario. Adhering to this figure, it can be easily understandable that, in any F2C-enabled smart city, a vast number of small areas may emerge to efficiently serve the F2C facilities near to the edge of the network. These small areas are individually known as fog areas. According to our defined F2C architecture, each fog area has two main resource components (i.e., control-area-unit (CAU) and Leader Fog nodes (LFn)), which are responsible for bringing the various cloud facilities (e.g., security and privacy functionalities, data storing, task allocation) near to the edge of the network. From Fig. 1, we can also quickly identify that many small computing devices are working in the fog area. These small computing devices are known as Normal Fog nodes (NFn) and are acting as the gateway for the most resource-constrained Edge-IoT devices (e.g., sensor nodes). The uppermost layer of our proposed architecture is known as the cloud layer. According to the expected functionalities, we have already classified the cloud layer resources into two parts; one is the F2C Controller, and the other one is the Cloud Agent (CIA). The F2C Controller is acting as the master node acting for providing certificate authority-based authentication for distributed CAUs [28], whereas the CIA is mainly in-charge of performing the high computational tasks and also responsible for efficiently managing the LFn resources. In [27] we already updated the initial version of the F2C architectural blocks by implementing the distributed database over all the CAUs and the F2C controller. That ensures to keep the data near to the edge of the network, which helps to provide better service experiences for latency-sensitive applications. Interestingly, keeping data near to the edge of the network eventually helps to perform some computational tasks at the edge, for rapidly forecasting the behaviour of system resources.

#### IV. A FRAMEWORK FOR RESOURCE PREDICTION MECHANISM IN F2C

For the sake of better understanding the prediction mechanism in the F2C paradigm, we have already depicted the thorough architectural framework in Fig. 2. The presented architecture is an initiatory approach for building a prediction-based resource management mechanism in the F2C paradigm. As the F2C is still in a development stage, there are many opportunities for improvement. For example, to build a prediction-

based resource allocation technique in the F2C paradigm, we have introduced many resource modules (i.e., Smart Cache, Task Scheduler) in our architecture, but confirming the final placement of these modules and the coordination among them is still a challenging job. Certainly, to improve the proposed architecture, that challenge (and many other) needs to be addressed.

We earlier described that, by performing computational tasks near to the edge of the network, the F2C is offering better service experiences for latency-sensitive applications and bringing more intelligence for any modern computing platform. So, considering this fact, in this work, we propose a modified F2C architectural schema for performing the forecasting process near to the edge of the network. According to our proposing architecture (Fig. 2), similarly like as cloud layer resources, also the fog layer resources (i.e., mainly LFn and CAU) are performing the forecasting process to predicting resource usage and execution time for upcoming computational tasks. Following Fig. 2, it can be easily understood that mainly LFn and CAU are the main two F2C resources; those are jointly performing the prediction process near to the end-users and managing all the NFn. So, considering this fact, we further classified the fog layer into two sub-layers; one is Fog Manager Layer (FML), and another one is Fog Employee Layer (FEL). All the FML sub-layer resources (i.e., LFn and CAU) are mainly in-charge for bringing the cloud facilities near to the edge of the network and managing all other FEL sub-layer resources (i.e., NFn). In this section by following the modified F2C architecture (Fig. 2), we are going to briefly describe the various F2C resource modules which are potentially involved in the forecasting/prediction process. Then, by briefly discussing the prediction/forecasting process in F2C paradigm, we are going to explain how the F2C resources can be adequately allocated for executing some tasks.

##### A. F2C resource modules: Involved for predicting resource usage and performance

We earlier described that the F2C is a hierarchical, combined and distributed computing platform. Interestingly, in the F2C paradigm, many resource modules are working together to achieve higher accuracy for predicting resource usage and execution time, what might lead to building an intelligent and proper resource management mechanism in the F2C paradigm. Next, we focus on those resource modules briefly.

1) *Smart Cache*: This is the core module, which is involved in forecasting/predicting resource usage and performance. It consists of two sub-modules *Training Module* and *Predictor Module*. The job of the *Training Module* is to build and train the machine learning (ML) model, whereas the *Predictor Module* is responsible for forecasting the resource usage and performance assisted by the trained model. In our proposed architecture, as the CIA and LFn are responsible for performing some computational tasks and controlling other F2C resources, so it is pretty much evident that the Smart Cache is an integral part of these two kinds of F2C resource components.

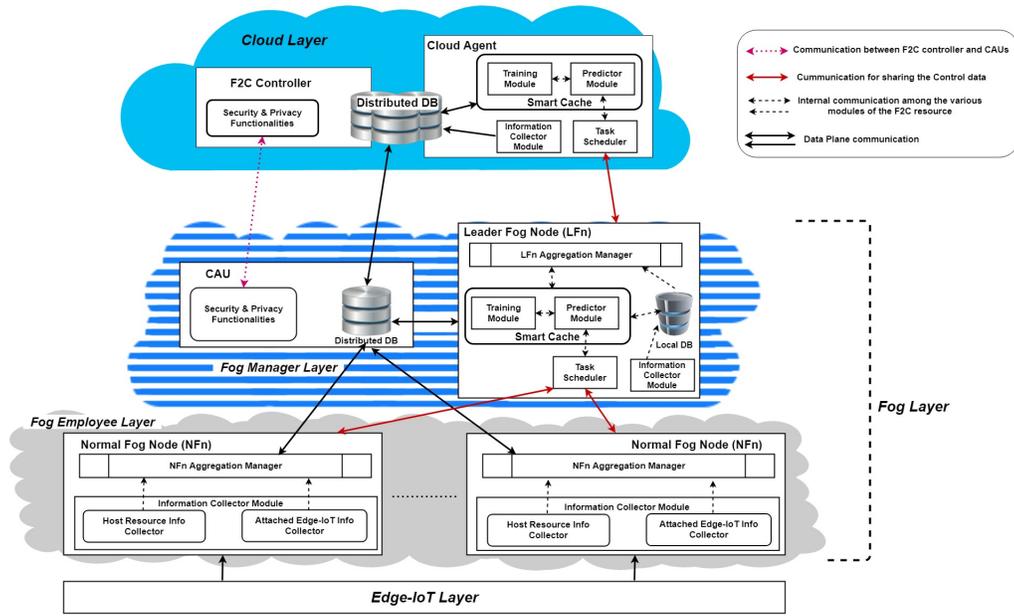


Fig. 2. Architectural framework for predicting the resource usage and performance in F2C paradigm

2) *Task Scheduler*: Based on the prediction result and following the requested computational task information, this module helps to choose and allocate the appropriate F2C resource for performing the task. Basically, it is a provisioning module, which helps to find out the suitable F2C resource for executing the requested task. Similarly, like as the Cache Module, it is also residing inside the CIA and LFn and assisting them to choose the proper F2C resources for executing some requested tasks.

3) *Aggregation Manager*: As the name suggested, this module is in-charge of aggregating the current state information and behavioural information of F2C resource and sending that aggregated information to the nearest distributed database node (i.e., CAU). It is also responsible for sending the captured or sensed data to the distributed database node. This resource module is an integral part of the LFn and NFn resources.

4) *Information Collector Module*: This module is a kind of monitoring module, which is responsible for continuously collecting the host resource information and also collecting the captured sensed data from the attached Edge-IoT devices.

Besides all of these resource modules, the fact of keeping and distributing the data near to the edge of the network (i.e., implementing the distributed database over the F2C controller and CAUs) will improve the prediction procedure, also eventually helping to speed-up the resource allocation procedure in the F2C paradigm.

#### B. Prediction-based resource management mechanism: Steps for forecasting and allocating the F2C resource

For any modern computing platform (i.e., fog computing) predicting the system resource usage and performance is the initial footprint to design an intelligent and proper resource

provisioning and management mechanism [29]. By considering this fact, in Fig. 3, we present a flow-chart diagram to show the process of prediction-based resource management mechanism in the F2C paradigm.

Following any smart computing platform and considering all the challenges for defining the proper and smart resource management mechanism of our proposed F2C paradigm, we understand that the type of target classes (e.g., RAM usage, CPU usage, task execution time) are known to us. Since the type of target classes for prediction are known, we adopted the *supervised* machine learning technique, to predict the resource usage and execution time for upcoming requested tasks.

Dataset is a vital ingredient for the machine learning (ML) technique. So before designing and training the ML model for forecasting future behaviour and states of the system resources, it is indispensable to either clean-up or process the available dataset and find out the relevant categories of data. More precisely, it is necessary to identify all the features which have an impact on the target variable. From the relevant set of existing techniques for selecting the relevant features, we adopted the *Pearson correlation coefficient* matrix [30] for identifying the relevant features, which have a high impact on the target classes (e.g., RAM usage, CPU usage, task execution time). The *Pearson correlation coefficient* ( $r_{xy}$ ) for some given paired data (e.g.,  $(x_1, y_1), \dots, (x_n, y_n)$ ) is described as follows:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

where  $n$  is the sample size;  $x_i, y_i$  are the individual sample points indexed with  $i$ ; and  $\bar{x}, \bar{y}$  are *sample mean*. In the proposed F2C architecture, as the data is distributed over the CAUs and the F2C Controller, it is relevant to perform the data

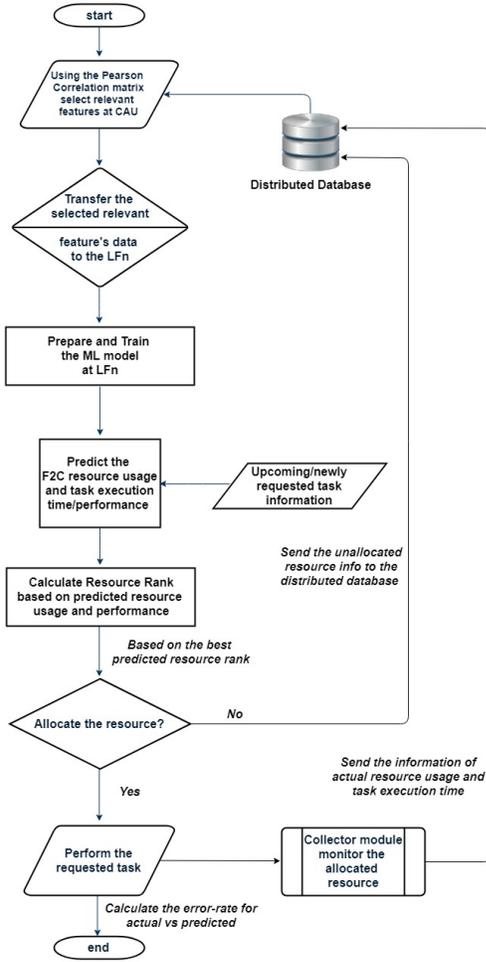


Fig. 3. Flowchart: Prediction-based resource management mechanism in F2C

filtering process in the CAUs and F2C Controller. After that, the filtered data is transferred to LFn and CIA for performing the prediction process. In our F2C paradigm, LFn and CIA both have a particular resource module (i.e., Smart Cache) for performing this prediction process.

After getting the filtered data, based on some existing ML technique, the *Smart Cache* (more precisely *Training Module*) build a trained ML model for predicting the upcoming/newly requested tasks. Then, based on this trained ML model, the *Predictor Module* forecasts both the system resource usage ( $Ru_p'$ ) and execution time ( $Ex_p'$ ) for the available F2C resources, in order to perform the newly requested task. Then, before allocating the newly requested task (considering the predicted resource usage ( $Ru_p'$ ) and execution time ( $Ex_p'$ )), the *Task Scheduler* module calculate the individual ranking (i.e.,  $R_r = (Ru_p', Ex_p')$ ) for all the available F2C resources and after that, it (i.e., *Task Scheduler*) chooses the best possible ranked ( $R_r$ ) resource for executing the newly requested task. Once the task has been allocated and performed, the monitoring module (as earlier described all F2C resources (i.e., CIA, LFn and NFn) have a special kind of monitoring module, referred to as *Information Collector Module*), collects

the actual resource usage and complete task execution time information. Then with the help of *Aggregation Manager*, it sends this information to the nearest node (i.e., CAU or F2C Controller) of the distributed database cluster. Significantly this information improves the accuracy level to choose the best-fitted resource for executing the next upcoming task.

## V. PERFORMANCE EVALUATION

Building the ML-based prediction model and improving its accuracy-level is the initial step for designing a proper and smart resource management mechanism in any computing paradigm. So, considering that fact in this work, we focus on building an accurate ML-based prediction model and evaluating its performance in our considering F2C domain. For that purpose, we already configured our project testbed [31] for performing some tests and evaluate the accuracy of our proposing ML-based mechanism. Then taking a vast amount of images, we execute some simple image-recognition application [32] in the F2C resources (except CAUs and F2C Controller), to generate the raw dataset<sup>1</sup> for predicting the resource usage and execution time in the F2C paradigm.

In our research lab, we have set up a prototype of the F2C-enabled smart city scenario, where the Cloud Layer elements (i.e., F2C Controller and Cloud Agent) are hosted on a server with Intel Xeon family E5-2620 V4 series (clock speed @3GHz), 96GB RAM, 1TB Hard Drive, and running on Ubuntu 16.04LTS Linux. The Fog Layer resources (i.e., CAUs, LFn and NFn) are relatively small computing devices, and following Fig. 1, it can be easily seen that the number of participating fog layer resources are quite higher than the cloud layer resources. Most importantly, they are hugely diverse [33]. For that reason and to avoid the unnecessary resource shortage issues, we created a significant number of VMs (approx.~ 32) in two relatively high-end computing machines. The specification of every VM is mostly different than others. Some VMs are relatively a bit resource-enriched, and some are not so. In our testbed, we considered those resource-enriched VMs as the FML sub-layer resources (i.e., CAUs and LFn) and except them, all others are considered as NFn resources. All fog layer resources are running on Ubuntu 16.04LTS Linux. We implemented a distributed database cluster over all the CAUs and F2C Controller. For that purpose, we are using the containerized Apache Cassandra (*Dockerized-Cassandra*). Tests have been performed implementing the multi-datacenter, multi-node based Cassandra cluster over the considered distributed framework.

From the observation of an smart city scenario, one can easily understand that at the edge of the network, the communication bandwidth is reduced. Considering this fact, in our proposed schema, we limited the maximum connection speed between Edge-IoT devices and the NFn to 2 MBps. Likewise, the maximum connection speed between NFn and LFn/CAUs is 5-7 MBps. Moreover, the maximum connection speed between different FML's architectural elements (e.g. LFn and

<sup>1</sup><https://gitlab.com/data-snoopers/appload>

CAUs) and also with the cloud layer’s element is 11 MBps. A thorough observation at any smart city scenario, quickly shows that bandwidth utilization is one of the most critical issues to manage. Considering this setup, we perform the tests to validate the effectiveness of our proposing architectural schema.

#### A. Dataset generation for train the ML model

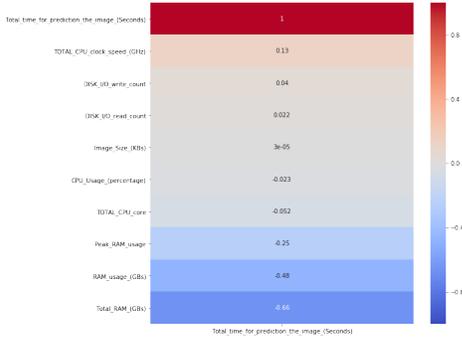


Fig. 4. Feature selection for Performance/Execution time prediction

We earlier identified that cleaning the raw dataset and filtering the appropriate data is the primary step for performing some ML technique. According to our proposed architecture, this process is done on CAUs or F2C Controller. As our main goal is to predict the performance of F2C resources, first we filter our collected raw dataset, using the *Pearson correlation coefficient* matrix to find out those features having a strong influence for predicting the task execution time. After filtering the raw dataset, we may conclude that in our experiment, task execution time is critically dependent on the RAM related information (i.e., Total RAM size and RAM usage) of the system resources. Also, it is minorly correlated with the CPU information (i.e., CPU clock speed) of F2C resources. In Fig. 4, we represented the Pearson correlation heatmap for showing the correlation of all other independent variables with the output variable: execution time (i.e., Total\_time\_for\_prediction\_the\_image\_(Seconds)). Following that figure (Fig. 4) and considering the general scaling of correlation values [34], it can be easily concluded that the task execution time has a moderate negative correlation with RAM features of F2C resources (i.e., Total RAM size and RAM usage). Also, the task execution time has a minor positive correlation with CPU information (i.e., CPU clock speed).

For any computing system the task execution time is strongly related with the system resources [35]. Hence, we can conclude that before designing an intelligent and proper resource management mechanism for the F2C system, it is pretty relevant to predict the resource usage for the available system resources, what would implicitly help to reduce the resource consumption of the whole F2C system. From the earlier experiment, as we found that the task execution time is significantly correlated with the RAM related information of an F2C resource; so we perform the same procedure (i.e.,

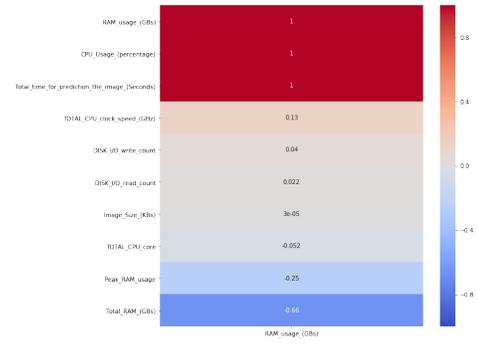


Fig. 5. Feature selection for RAM usage prediction

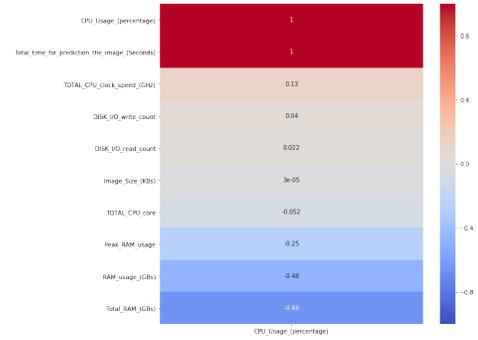


Fig. 6. Feature selection for CPU usage prediction

*Pearson correlation coefficient*) to identify the features which have strongly correlated with RAM usage information. In Fig. 5, we presented the heatmap to understand which features are strongly correlated with RAM usage. Interestingly, we found that, CPU usage and task execution time both have a very high positive correlation with the RAM usage of F2C resources. So, then we also follow the same procedure to identify all the features; those have a strong correlation with the CPU usage of F2C resources. In Fig. 6, we depicted the heatmap to identify all the features which correlate with the CPU usage of F2C resources.

Surprisingly we found that, the Storage information (i.e., Disk I/O) of F2C resources, has not any significant correlation with the task execution time, RAM usage and CPU usage information. So then we assume that it is worthily to predict the RAM usage, CPU usage and task execution time for designing a proper and intelligent resource management mechanism for the F2C system. To that end, according to our proposed F2C model, after this filtering process, the cleaned and filtered data must be transferred/copied to the LFn or CIA for further processing.

#### B. Result: Resource usage and performance prediction

Following the F2C paradigm, we understand that, due to the resource-constrained nature of fog layer resources, it is very much tough for them to perform some additional heavy computation work for prediction. For that reason, it is necessary to choose an appropriate and low computation complexity-based

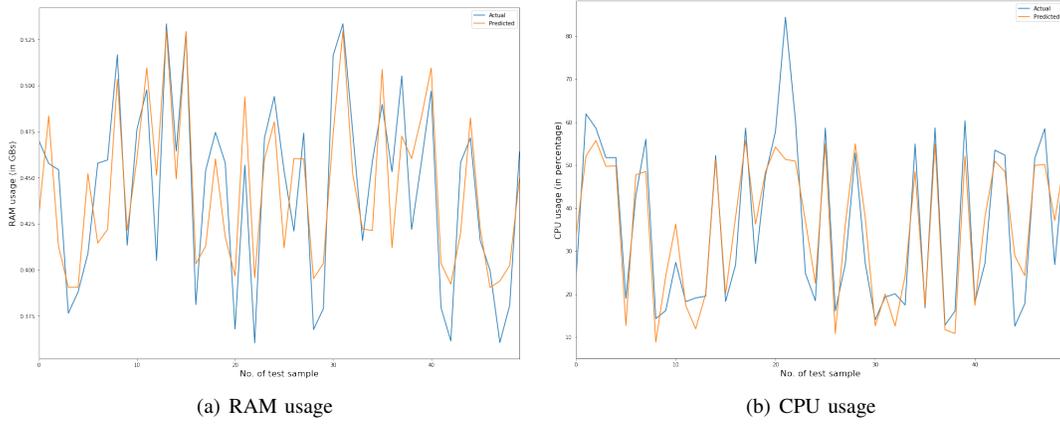


Fig. 7. Model's performance evaluation: Actual vs Predicted

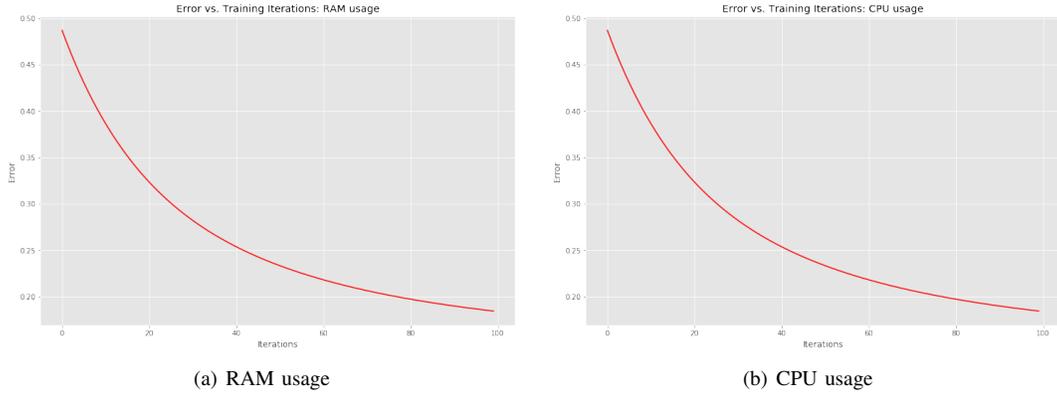


Fig. 8. Accuracy evaluation in terms of cost function value: Error vs Training iteration

ML technique for our F2C paradigm. As the linear regression ensures to consume less amount of resources for performing [36]; so to predict the resource usage and performance of the F2C resources, we adopted the *multivariate linear regression* ML technique [37]. Based on this ML techniques, we build and train the ML model for predicting the resource usage and execution time in our F2C paradigm. Importantly, evaluating the accuracy of the prediction model is highly necessary to justify the effectiveness of our proposed architectural schema. For that purpose, we calculate cost function value ( $J(\Theta)$ ) to evaluate the prediction quality, as follows:

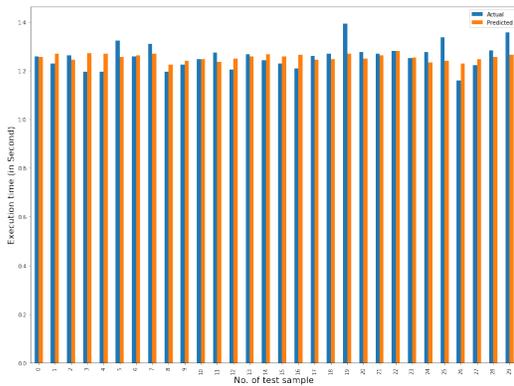
$$J(\Theta) = J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (2)$$

Where  $m$  is the number of sample data,  $h_{\theta}(x)$  is the hypothetical value, which has been predicted by our ML model and  $y$  is the actual value. Now, to build more accurate prediction model for our F2C paradigm, we adopted the *gradient descent* algorithm [38] to reduce the value of the cost function ( $J(\Theta)$ ).

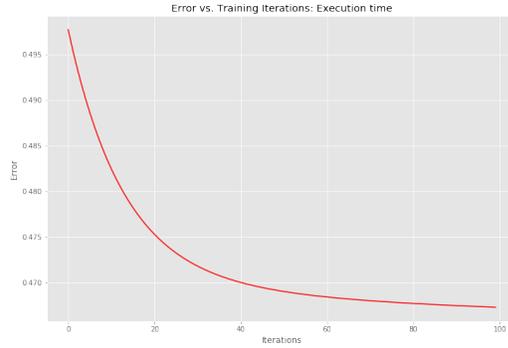
It has been earlier mentioned that before choosing the appropriate resource for executing the newly requested task, it is pretty relevant to predict/forecast the execution time for the newly requested task, and resource usage (e.g., RAM and CPU usage) of the available F2C resources. For that purpose,

we already built and trained our ML model for predicting the execution time and future resource usage. Then, we perform some tests to validate the efficiency of our model. In Fig. 7(a), 7(b) & 9(a), we present the test results (i.e., predicted vs actual) to show the accuracy of our ML model for predicting the resource usage and future task execution time in F2C paradigm.

During our experiments, we found that initially, the cost function values of our ML model were a bit higher. Initially the difference values for all the three cases (e.g., RAM usage, CPU usage and execution time prediction) were approx.  $\sim 4\%$ ,  $13\%$  &  $8\%$  respectively. So then, to reduce the error (i.e., cost function value) and build a more accurate prediction model, we adopted the *Multivariate Gradient Descent* algorithm to train the predictive functions of our ML model (i.e., more precisely, the theta ( $\Theta$ ) parameters of the hypothesis function ( $h_{\theta}(x)$ )). After performing some iteration of training, we have seen that our training model can accurately predict the resource usage (e.g., RAM usage and CPU usage) and also the execution time for the F2C resources. In Fig. 8(a), 8(b), & 9(b), we represent the accuracy evaluation for our prediction model. We found that for all three cases (i.e., RAM usage, CPU usage and execution time) almost after hundred (100) iterations of training, our ML model is able to accurately perform and



(a) Actual vs Predicted



(b) Accuracy evaluation

Fig. 9. Model’s performance measurement and accuracy evaluation: for predicting execution time

reduce all the three cost function values<sup>2</sup> (e.g., for RAM usage, CPU usage, execution time) to 1.8%, 7% & 4.6% respectively.

By performing all these operations, we ensure the accuracy of our predicting model in the F2C paradigm. Most importantly, an accurate prediction helps to choose the best-fit resources for the upcoming tasks and also eventually helps to design a proper resource management mechanism for the F2C paradigm. Surprisingly, in our experiment, we have found that the disk I/O information has not too much impact on predicting the execution time. We generated the raw dataset by executing a single type of application (e.g., image-recognition), so it might happen that, for executing this application the disk I/O is not so influencing for changing the task execution time. So, in future, it is necessary to execute other kinds of applications and investigate the correlational value between the task execution time and disk I/O. As this work is the initial step to building the prediction based resource management mechanism in the F2C paradigm, we assumed that measuring and predicting the network I/O is out of scope for this work.

## VI. FUTURE WORK & CONCLUSION

This paper presents an architectural schema for forecasting resource usage and performance (i.e., task execution time) prediction in combined F2C systems, based on standard supervised machine learning technology. The proposed architectural schema is the primary footstep to pave a new path for any F2C-based smart computing system. Most importantly, the prediction of resource usage and performance (i.e., task execution time) helps to find out the appropriate set of resources to be allocated for executing upcoming computational tasks. Implicitly, following that way, we can also ensure better resource utilization in a F2C-based computing system. In the proposed experiments, we have found that after training our model for hundred iterations, the difference between the predicted and actual measurement for all three experiments (e.g., RAM usage, CPU usage and execution time prediction) are becoming low. Initially the difference values for all the three cases (e.g., RAM usage, CPU usage and execution time

prediction) were approx.  $\sim 4\%$ ,  $13\%$  &  $8\%$  respectively; but after training our model for hundred iterations the differences goes down to,  $1.8\%$ ,  $7\%$  &  $4.6\%$  respectively.

To validate our architectural schema, we have performed some tests to predict the resource usage and its performance. As the target classes (i.e., RAM usage, CPU usage, execution time) for prediction are known for us, in this work we adopt some existing supervised machine learning technique to forecast the F2C resource usage and performance. However, as we mentioned earlier, the F2C is still in its infancy and even this work is the initial step for making an efficient and intelligent resource provisioning and management mechanism, so many challenges remain to be addressed. For that reason, some additional work is essential. For example, in this work, we perform all tests considering a single type of application (image recognition), but in a real scenario, any F2C-based computing system should be capable for executing various types of applications to provide different services (e.g., e-Health facilities, traffic management, etc.). So, in that case, it is also really essential to measure the performance of our proposed schema. Importantly, according to the rank ( $R_r$ ) of F2C resources, we are allocating them for executing some tasks. So, it is also necessary to compare the performance of our prediction-based resource management approach with other existing resource management approaches. However, we have not found any such substantial work in F2C domain to be compared with our proposal. Thus, this work presents a preparatory approach for designing an appropriate and intelligent resource management mechanism in the F2C domain, which has been planned to lay the groundwork for a more complete research study in the future.

## ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Science, Innovation and Universities and the European Regional Development Fund (FEDER) under contract RTI2018-094532-B-I00, and by the H2020 European Union mF2C project with reference 730929.

<sup>2</sup><https://github.com/resourceusageds/prediction>

## REFERENCES

- [1] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, 2016.
- [2] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.
- [3] A. Suresh and R. Varatharajan, "Competent resource provisioning and distribution techniques for cloud computing environment," *Cluster Computing*, pp. 1–8, 2017.
- [4] N. Jain and I. Menache, "Resource management for cloud computing platforms," Sep. 20 2016, uS Patent 9,450,838.
- [5] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource provisioning for iot services in the fog," in *2016 IEEE 9th international conference on service-oriented computing and applications (SOCA)*. IEEE, 2016, pp. 32–39.
- [6] P.-O. Östberg, J. Byrne, P. Casari, P. Eardley, A. F. Anta, J. Forsman, J. Kennedy, T. Le Duc, M. N. Marino, R. Loomba *et al.*, "Reliable capacity provisioning for distributed cloud/edge/fog computing applications," in *2017 European conference on networks and communications (EuCNC)*. IEEE, 2017, pp. 1–6.
- [7] K. Ahmed, D. Bigagli, Z. Hu, and J. Wang, "Resource manager for managing the sharing of resources among multiple workloads in a distributed computing environment," Apr. 25 2017, uS Patent 9,632,827.
- [8] G. Galante, L. C. E. De Bona, A. R. Mury, B. Schulze, and R. da Rosa Righi, "An analysis of public clouds elasticity in the execution of scientific applications: a survey," *Journal of Grid Computing*, vol. 14, no. 2, pp. 193–216, 2016.
- [9] W. Zirwas, L. Thiele, T. Wirth, and T. Haustein, "Controlling scheduling decisions in a distributed cooperation system," Jan. 10 2017, uS Patent 9,544,913.
- [10] A. A. Rahmanian, M. Ghobaei-Arani, and S. Tofighy, "A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment," *Future Generation Computer Systems*, vol. 79, pp. 54–71, 2018.
- [11] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.
- [12] Z. Cao, J. Lin, C. Wan, Y. Song, Y. Zhang, and X. Wang, "Optimal cloud computing resource allocation for demand side management in smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1943–1955, 2016.
- [13] J. S. Hirwa, U. Rugwiro, M. Stammers, and C. Gu, "Cloud-based clusters: Multivariate optimization techniques for resource prediction," in *2016 15th International Symposium on Parallel and Distributed Computing (ISPDC)*. IEEE, 2016, pp. 165–171.
- [14] T. Mehmood, S. Latif, and S. Malik, "Prediction of cloud computing resource utilization," in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*. IEEE, 2018, pp. 38–42.
- [15] S. Kumaraswamy and M. K. Nair, "Intelligent vms prediction in cloud computing environment," in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. IEEE, 2017, pp. 288–294.
- [16] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Efficient datacenter resource utilization through cloud resource overcommitment," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, pp. 330–335.
- [17] H. A. Chaudhari, C. S. Thaker, and J. Patel, "Performance improvisation through resource usage optimization in cloud environment," in *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*. ACM, 2014, p. 45.
- [18] B. P. Hayes and M. Prodanovic, "State forecasting and operational planning for distribution network energy management systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1002–1011, 2015.
- [19] S. S. Chaudhari and R. C. Biradar, "Resource prediction using wavelet neural network in mobile ad-hoc networks," in *2014 International Conference on Advances in Electronics Computers and Communications*. IEEE, 2014, pp. 1–6.
- [20] M. Aazam, M. St-Hilaire, C.-H. Lung, I. Lambadaris, and E.-N. Huh, "Iot resource estimation challenges and modeling in fog," in *Fog Computing in the Internet of Things*. Springer, 2018, pp. 17–31.
- [21] C. N. Le Tan, C. Klein, and E. Elmroth, "Location-aware load prediction in edge data centers," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 25–31.
- [22] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops)*. IEEE, 2015, pp. 105–110.
- [23] F. Bi, S. Stein, E. Gerding, N. Jennings, and T. La Porta, "A truthful online mechanism for allocating fog computing resources," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1829–1831.
- [24] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–6.
- [25] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688–1699, 2017.
- [26] H. Xu, H. Huang, R. S. Khalid, and H. Yu, "Distributed machine learning based smart-grid energy management with occupant cognition," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2016, pp. 491–496.
- [27] S. Sengupta, S. Kahvazadeh, X. Masip-Bruin, and J. Garcia, "SFDDM: a secure distributed database management in combined Fog-to-Cloud systems," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (IEEE CAMAD 2019) (in press)*, Limassol, Cyprus, Sep. 2019.
- [28] S. Kahvazadeh, V. B. Souza, X. Masip-Bruin, E. Marín-Tordera, J. Garcia, and R. Diaz, "Securing combined fog-to-cloud system through sdn approach," in *Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms*. ACM, 2017, p. 2.
- [29] S. Agarwal, S. Yadav, and A. K. Yadav, "An efficient architecture and algorithm for resource provisioning in fog computing," *International Journal of Information Engineering and Electronic Business*, vol. 8, no. 1, p. 48, 2016.
- [30] H. Xu and Y. Deng, "Dependent evidence combination based on shearman coefficient and pearson coefficient," *IEEE Access*, vol. 6, pp. 11 634–11 640, 2017.
- [31] X. Masip-Bruin, E. Marín-Tordera, A. Juan-Ferrer, A. Queralt, A. Jukan, J. Garcia, D. Lezzi, J. Jensen, C. Cordeiro, A. Leckey *et al.*, "mf2c: towards a coordinated management of the iot-fog-cloud continuum," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*. ACM, 2018, p. 8.
- [32] M. Olafenwa, J. Olafenwa, and K. vcar, "Olafenwamoses/imageai," Aug 2019. [Online]. Available: <https://github.com/OlafenwaMoses/ImageAI>
- [33] S. Sengupta, J. Garcia, and X. Masip-Bruin, "Taxonomy and resource modeling in combined fog-to-cloud systems," in *Proceedings of the Future Technologies Conference*. Springer, 2018, pp. 687–704.
- [34] D. E. Hinkle, W. Wiersma, and S. G. Jurs, *Applied statistics for the behavioral sciences*. Houghton Mifflin College Division, 2003, vol. 663.
- [35] C. Delimitrou and C. Kozyrakis, "Quasar: resource-efficient and qos-aware cluster management," in *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1. ACM, 2014, pp. 127–144.
- [36] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2016, pp. 1310–1315.
- [37] R. Singh, R. K. Umrao, M. Ahmad, M. Ansari, L. Sharma, and T. Singh, "Prediction of geomechanical parameters using soft computing and multiple regression approach," *Measurement*, vol. 99, pp. 108–119, 2017.
- [38] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.