National College of Ireland

DynamicForecast: Experts Council with optimized Workload Prediction Framework for Cloud Computing

MSc Cloud Computing
Research Project

Kripa Mariam Joy
Student ID: 20217986

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Kripa Mariam Joy
Student ID:	20217986
Programme:	MSc. Cloud Computing **Year:** 2022
Module:	Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	15/08/2022
Project Title:	DynamicForecast: Experts Council with optimized Workload Prediction Framework for Cloud Computing
Word Count:	……5618…… **Page Count**……19……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: *KripaMJ*

Date: 01/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DynamicForecast: Experts Council with optimized Workload Prediction Framework for Cloud Computing

Kripa Mariam Joy
20217986

Abstract

Resource management in cloud environment is a challenging task. Management of resources with predictive scaling technique has been used to circumvent the limitation of reactive scaling in the cloud such as over-provisioning and under-provisioning of resources. The predictive technique aids in workload (WL) predictors, which predict the fluctuations in workload. However, the accuracy of predictors varies as per the varied workload pattern. In order to address this, a novel workload prediction framework naming DynamicForecast (DF) has been introduced. DF works by using a stack of predictors with long-short-term memory (LSTM) model along with Adam's optimization technique to build the ensemble models which further predicts the workload more precisely. The performance is measured by Mean Absolute Percentage Error (MAPE). As MAPE decreases the accuracy of prediction increases. DF has 50% - 94% lower MAPE compared with state-of-the-art predictors (CloudInsight and ARIMA) for WLs which are bursty, random, and seasonal.

Keywords - Cloud Computing; WL Prediction; Resource Management; Long Short-Term Memory; Machine Learning; Ensemble Model

1 Introduction

Cloud computing is widely adopted from past decade due to its number of characteristics as per NIST [8]. The cloud service providers (CSP) always ensure quality of service (QoS) with high elasticity and cost-efficient way to its potential consumers [1]. With the aid of autoscaling, elasticity can be achieved in cloud. Due to reactive approach of resource allocation in cloud environment, it leads to over-provisioning and under-provisioning issues. The exhaustion of cloud resources takes place in case of over-provisioning. However, in case of under-provisioning, along with resources exhaustion, service level agreements i.e., SLAs are also violated. In order to provide solutions to these issues, predictive approach is used which makes use of accurate workload prediction to allocate resources in an efficient manner.

Workload forecast could be done using multiple methods such as statistical time series method, machine learning method, deep learning approach and so on [3]. Even though there are dynamic fluctuation in pattern, the existing predictive approach have an assumption that the pattern of workload is same for specific time and predictors were developed for specific load pattern. Therefore, prediction of accurate workload has become a hectic task and still considered as an open issue for fluctuating load pattern [4] due to the dynamic

behaviour of WL pattern in accordance with time. Figure 1 illustrates the dynamic nature of WL pattern. The WL patterns drastically vary among the different applications in cloud.

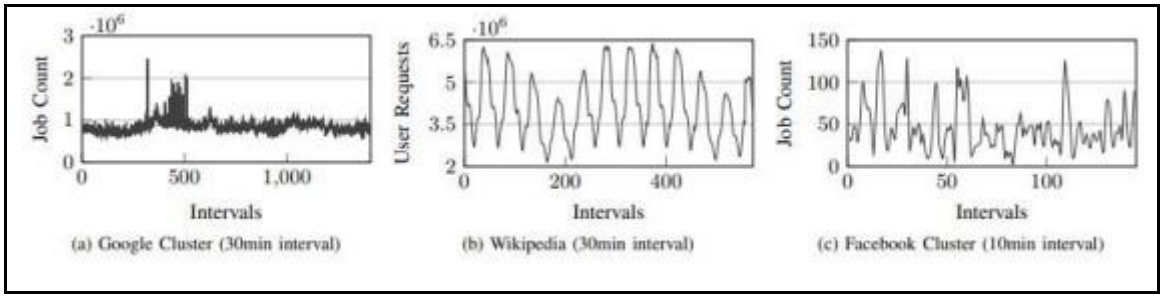

Figure 1: Traces of three workload with multiple patterns [2]

The patterns of WLs such as increasing WL, On and Off WL, cyclic Bursty and Random WL [14] [15] pattern needs to be analysed thoroughly for the accurate prediction of WL and to find out the perfect predictors for each pattern. It is obvious that there is no single WL predictor which fits for every pattern by analysing 21 conventional prediction algorithms which includes 2 native algorithm, 6 regression algorithm, 7 time-series and 6 machine learning algorithms [4]. The variation of accuracy for predictors according to multiple patterns are demonstrated in Figure 2. So, it is needed to have a universal framework for workload prediction, that can produce more precise prediction for various dynamic workloads.

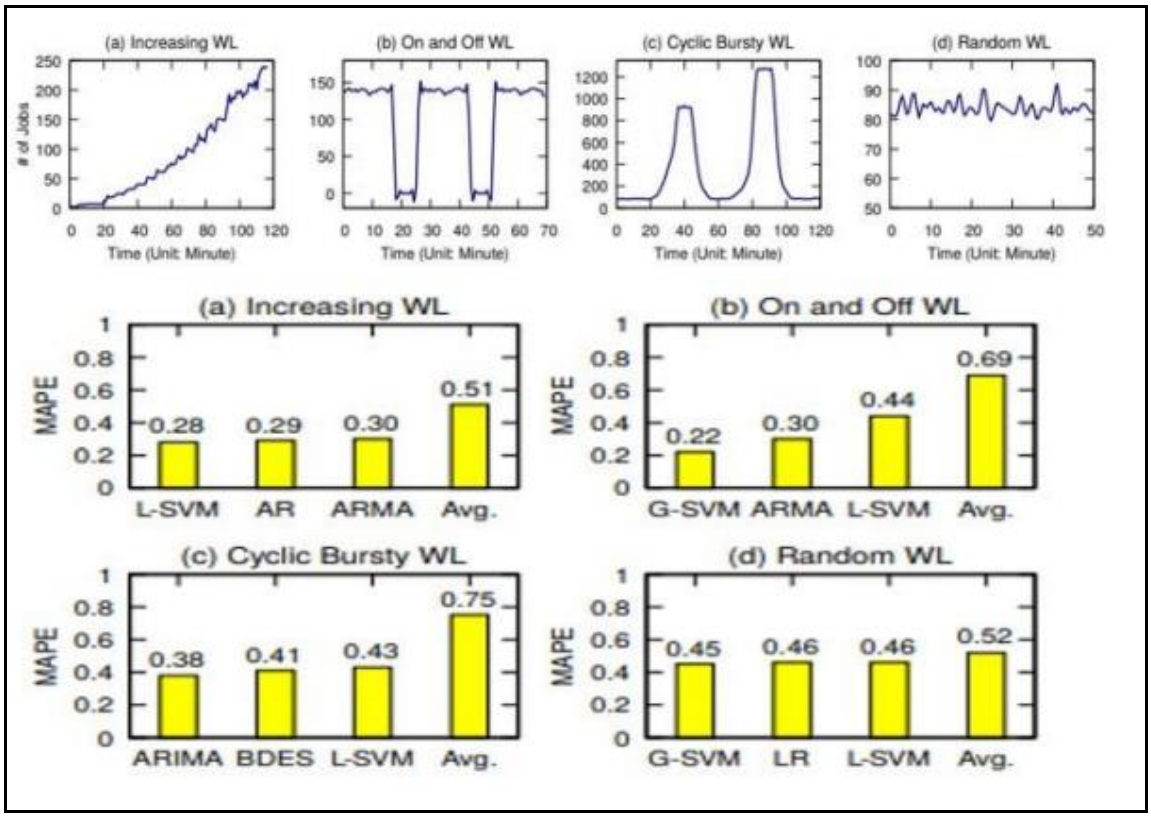

Figure 2: Workload patterns and MAPE of different predictors for multiple patterns [2]

Therefore, the research question is phrased as “Can the precision of WL prediction be improved for efficient resource allocation with DF, using the novel general WL prediction framework with a pool of predictors using machine learning models and deep learning model, LSTM along with Adam's optimization technique?”

DynamicForecast (DF) is a novel framework for cloud WL prediction which have empirical structure consists of stack of predictors to analyse future job. Because there is no individual finest predictor for varying workload patterns [4]. Then the predictor pool of machine learning models is incorporated with deep learning model, that is; LSTM model along with an effective optimization method called Adam's optimization [2]. The output of these predictors is then stored in a repository to create ensemble models and for future prediction of WL more precisely.

The further sections of this paper is structured as follows. Section 2 summarizes the related work. Section 3 describes the framework details. This is followed by the design specification and implementation in Section 4 and Section 5 respectively. In Section 6, evaluation of results obtained for DF with the existing frameworks. Section 7 provides conclusion and future works of DF.

2 Related Work

The section “Related Work” delivers an outline of similar research works of literature within the same technical area. This covers a broad range of techniques recommended for forecasting or predicting the workload, previous works, and so on. The analysis, evaluation, and issues of those papers would be further discussed.

2.1 Prediction Systems

The general approaches used for prediction are machine learning, deep learning, regression, time series and so on. Even though numerous research studies exist for prediction, works held for forecasting WL and thereby managing resources efficiently are limited. LoadDynamics, PRESS, CloudScale, and SDWF are some successful outcomes of research for prediction of WL. This section illustrates the mentioned prediction system for workload in order to avoid issues with allocating resources in the cloud.

Cloud systems should have elastic resource allocation to bring down the cost of allocating resources while meeting Service Level Objectives (SLOs). The two-research works namely PRESS and CloudScale for elastic resource allocation are proposed in [6] and [7] in the year 2010 and 2011 respectively. To proactively controls resource provisioning and to recognizes fine grained dynamic patterns of resource requirement in applications, a new scheme for cloud based system is introduced called PRedictive Elastic reSource Scaling (PRESS. Light-weight signal processing and statistical learning algorithms are used to enable online forecasting of dynamic application resource requirements [6]. On the other hand, CloudScale automates the fine-grained elastic resource scalability of multi-tenant cloud computing infrastructures. This method utilizes online resource demand forecasting and prediction error handling to accomplish adaptive resource prediction without the need for prior knowledge of cloud applications. CloudScale is a migration-based scaling solution for

application which have scaling conflicts such as dynamic CPU voltage/frequency scaling for energy conservations with less impact on application SLOs [7]. Xen is used in the implementation of both PRESS and CloudScale. The system PRESS is based on Xen and has been examined using the benchmark called RUBiS and Google application load traces. They claim good precision in resource forecasting, with less than 6% overestimation and approximate zero underestimation [7]. Unfortunately, the PRESS prediction aids signal processing technique. Because of the complexity of the calculation, some delays are included in receiving the prediction output. Furthermore, for some signal processing techniques, the design area and power consumed are comparatively high. “Discrete Time Fourier Transform (DTFT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Decimation In Time-FFT (DIT-FFT), Decimation In Frequency-FFT (DIF-FFT)”, and so on are the multiple signal processing algorithms. Fast Fourier Transform signal processing is used in PRESS because it requires fewer additions and multiplications than other signal processing algorithms. Even though FFT algorithm use "divide and conquer" strategy to process the signal, FFT requires input signal as power of two. Else it affects the processing and accuracy of the output. PRESS is evaluated for close use cases, as it just examines a small sample of current resource needs to predict future demand [6]. However, CloudScale was built on top of Xen, and extensive evaluation was carried out using a set of CPU and memory-intensive applications generated by real Web server traces, Hadoop MapReduce systems, and a commercial stream processing system. The finding demonstrates that, compared to other competing systems, CloudScale can achieve significantly higher SLO conformance at a reduced energy and resource cost. CloudScale is lightweight and non-intrusive, with little impact on the virtualized computer cluster. While compared with other methods, about 82% less scaling issues can be resolved. Furthermore, with negligible impact on application performance and SLO compliance, this can save 8-10% total energy usage and 38-70% WL energy consumption. By abstracting the duplicating pattern in the resource utilization trace, long term conflict forecast is accomplished in CloudScale, which is an excellent topic. When a repeating pattern cannot be found, CloudScale employs a multi-step Markov prediction algorithm to make long-term predictions. However, multistep Markov prediction has limited forecast accuracy because the link between the resource forecast model and real resource demand weakens in the future [7].

Post introduction and use of machine learning and deep learning technique, the prediction of data achieved new heights. Because of LSTM's ability to predict time series, LSTM or LSTM-variants are broadly used. Deep learning models, such as LSTM, are recommended for forecasting future workloads. Though, a significant percentage of (workloads) training datasets and computer resources are required for the prediction. Two prominent research works that employ the deep learning technique are Self-Directed Workload Forecasting (SDWF) and LoadDynamics. The papers [2] and [4] respectively describes SDWF and LoadDynamics for predicting workload accurately. LoadDynamics is an innovative generic framework for workload prediction that makes use of LSTM models which automatically optimizes its internal parameters for each job. LoadDynamics is tested using a combination of WL traces that represent public cloud apps, scientific apps, data center jobs, and online applications. The test outputs reveal that LoadDynamics has average forecast error of 19%, which is at least 7% less than existing WL forecast methods. Also, the

best predictor found by exhaustive search for every WL has 1% higher error in LoadDynamics. Unfortunately, LoadDynamics has some constraints. The LSTM model used in LoadDynamics was narrowly trained from limited WLs, resulting in overfitting to the training dataset and a scarcity of detailed analysis with real-world traces. Moreover, it is ambiguous how the methodologies for tuning/updating the model hyperparameters, which frequently impact precision of the forecast [2]. In SDWF, the trend of forecast error is tracked by computing the divergence in previous prediction and used to enhance predictive modelling efficiency. The model provides a superior heuristic strategy focused on blackhole phenomena for neuron training. Six actual data traces from various settings are used to assess the effectiveness of the proposed strategy. The precision of the model is compared to existing models that use cutting-edge methodologies such as deep learning, differential evolution, and back propagation. The mean squared error of prediction is lowered up to 95% when compared to previous approaches. The statistical analysis further employs the Friedman and Wilcoxon signed rank tests to assess the efficacy of the proposed prediction model [5].

2.2 Ensemble Approaches

Ensemble prediction methods use variety of predictors to achieve high generalizability and improve performance. To minimize variance (bagging) or bias (boosting) on forecasting accuracy, ensemble approaches utilize bagging or boosting methodologies. ASAP and VADARA are two instances of ensemble approaches that are compared and contrasted in this section [9][10].

An online temporal data mining model is introduced to forecast and model Virtual Machine cloud demands called Self-Adaptive Prediction System (ASAP). To derive enhanced features from the VM provisioning request stream and notify the provisioning system, allowing VMs to be prepared in advance are the primary aim of ASAP. In order to address the quantification issue, they provide Cloud Prediction Cost, which encloses the cloud's cost and bounds and guides the training of forecasting algorithms. A two level ensemble method is used in ASAP to capture the characteristics of time series with high transient demands. As per the results based on recorded data from an IBM cloud in operation, the ASAP system significantly improves cloud service quality and permits on-the-fly provisioning [9]. Vadara is a fully general elasticity framework that incorporates and encloses the API behaviour of multiple Cloud Providers (CP). It presented a padding system for both under and over-provisioning based on the most recent prediction errors. Also, it uses the combination of several prediction methodologies together to forecast the workload. KNN is the load prediction algorithm used here [10]. These two techniques, such as ASAP and Vadara, concentrate on their analyses on a simple assumption that the most recent best predictors (e.g., the lowest cumulative error over the most recent monitoring interval) would perform best in the near future. However, it should be noted that this theory is not always correct. This idea can be countered by workloads that exhibit a significant degree of short-term burstiness.

2.3 Time series predictor approach

“Auto-Regression (AR), AutoRegressive Moving Average (ARMA), AutoRegressive Integrated Moving Average (ARIMA)” are examples of time series predictors. Among these predictors, ARIMA is most efficient predictor as it the combination of AR and ARMA. This section describes about research of ARIMA predictor.

One of the primary factors influencing QoS is dynamic workload behaviour, which results in variable resource requirements. Whenever the job arrival rate surpasses the capacity of resource exists, then degradation of QoS happens and also affect clients with application. In order to handle this issue, a proactive solution for provisioning the resources dynamically for SaaS services based on forecasts using the AutoRegressive Integrated Moving Average (ARIMA) model is proposed. This article assessed the precision of future WL prediction using real-world requests to web servers. The impact of achieved accuracy on resource utilization efficiency and Quality of Service (QoS) is evaluated. The result obtained from simulation of the model indicates that ARIMA has an average accuracy of 91%. In epitome, ARIMA is an extension of ARMA that combines AR and MA models to predict nonstationary time series data accurately. ARIMA is denoted by the letters p, q, and d, which represent the AR, MA, and differencing model orders, respectively. ARIMA is ideal for cyclic bursty workloads with strong trends and cyclic changes [16].

2.4 Multi-predictor Approach

To improve accuracy, Workload Classification and Forecast (WCF), a multipredictor technique is proposed in [12]. This system identifies and summarizes the traits and components of workload intensity characteristics for automatic classification and selection of prediction techniques. This technique classifies the intensity of workload behaviours in order to choose appropriate prediction techniques dynamically. This was achieved by utilizing direct feedback mechanisms which evaluate as well as investigate the current multiple prediction accuracy. They have been incorporated into a decision tree that takes into account of user-defined criteria of prediction objectives. This permits online application and processing of continuous prediction results for a wide range of WIBs. The real time traces are used to test this system [12]. However, in [13], an adaptive approach for predicting WL is proposed. This method categorizes WLs into various classes, which are then automatically assigned to multiple prediction models based on load features. The issue for WL classification is then converted into a task assignment problem by building a mixed 0-1 integer programming model and providing an online solution. Google Cluster traces are used to test this approach.

2.5 Combination of prediction system and ensemble approach

The prediction system as well as ensemble approach has its own benefits. Kim et al. in 2020 and 2018 use combination of these two concepts to propose CloudInsight in [1] [4]. CloudInsight has a predictor pool which consist time series predictors, machine learning predictors and so on. These predictors would be selected by the ensemble model created

through some historic data and choose the appropriate predictor according to the WL. CloudInsight address the issue of dynamic fluctuation of WL pattern. Even though, the accuracy of cloudInsight for the WL pattern such as random and bursty is less. Also, the predictors used in the pool are not so efficient compared with other existing predictors.

DF utilize the idea of cloudInsight as it addresses the main issue in forecasting the WL, that is dynamic fluctuation of WL pattern. The predictor pool is modified with machine learning models as well as LSTM model along with Adam's optimization. DF create four perfect ensemble models and prediction would be carried out.

3 Research Methodology

Research methodology consist several steps namely, data gathering, pre-processing, ensemble model creation, prediction and storage of result in repository.

Step 1: Data gathering includes the finding right dataset of cloud system with different WL pattern to analyse the proposed system.

Step 2: Pre-processing of data includes filtering of unwanted data from the dataset. Here the dataset selected are from grid WL[1]. This consist of various WL patterns from different sources. The datasets are pre-processed by eliminating the contents other than number of jobs arriving to the system according to the time.

Step 3: The next step involves splitting the dataset for training and testing and then creation of models for each predictor. This is followed by creation of four ensemble models according to the MAPE value. Among four ensemble models, most precise one is selected for prediction. Here 20% of data from dataset is selected for testing and rest for the prediction. The dataset is divided into 30 windows and each window has 25 predictions. The MAPE value obtained for each set of windows is compared and then create a set of four ensemble models with lower MAPE value.

Step 4: Post creation of ensemble model, the appropriate predictor would be selected from the set of four ensemble model and MAPE is calculated. These results are stored in a repository for the further prediction. In order to analyse the result easily, a GUI unit is coded with WL prediction using tkniter, which shows the MAPE value of the prediction. Figure 3 depicts the research methodology.

[1] http://gwa.ewi.tudelft.nl/datasets/

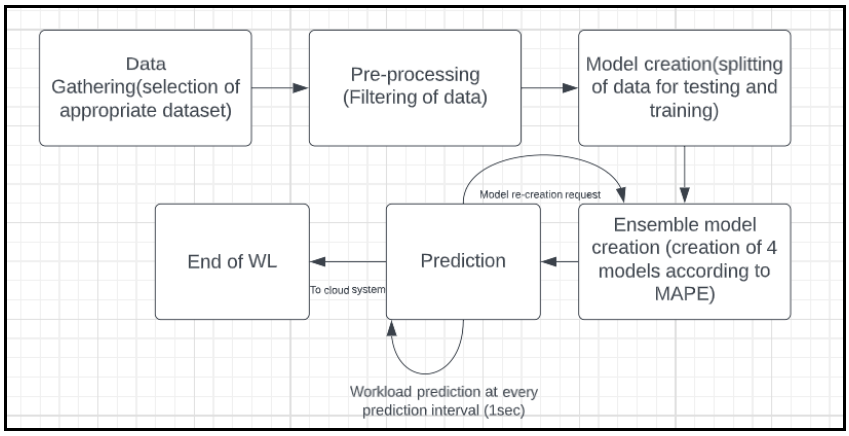

Figure 3: Research methodology of DF

4 Design Specification

The DF framework consist of predictor pool, ensemble model builder, workload repository, DF predictor as main components. The figure 4 depicts the framework of DF. The number of jobs arriving in time is considered as the input of the framework, DF. The prediction for near future WL is considered as the output. A stack of WL predictors is included in predictor pool along with LSTM model and Adam's optimization. The prediction history of predictors and job history are stored in WL repository. Post evaluation of each predictor's performance, ensemble model is created. This ensemble model aids in the prediction of future WL and the resource management component use this prediction for resource scaling.

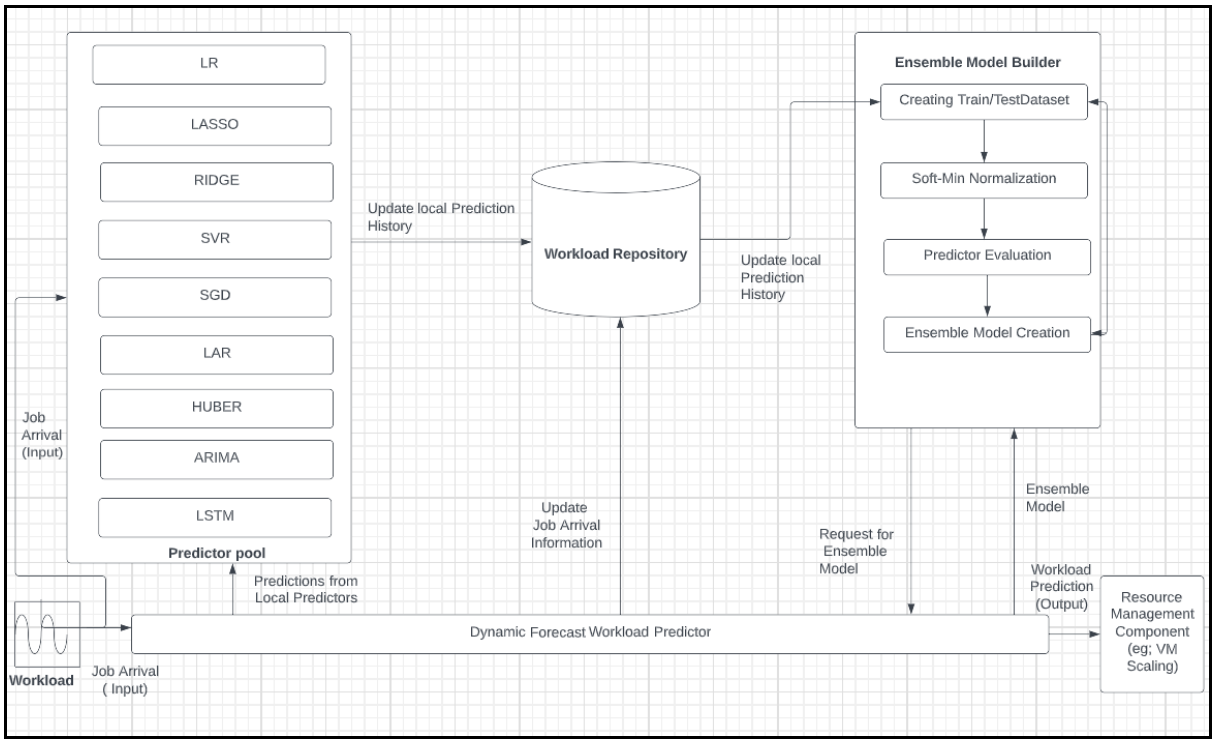

Figure 4: DynamicForecast framework

4.1 Predictor pool

Predictor pool consist of 8 machine learning models/predictors and one deep learning model called LSTM with Adam's optimization. Figure 5 illustrates predictor pool. The predictors included in predictor pool are linear regression (LR), Least Absolute Shrinkage and Selection Operator (LASSO), RIDGE, Support Vector Regression (SVR), Stochastic Gradient Descent (SGD), Least Angle Regression (LAR), HUBER, Auto Regressive Integrated Moving Average (ARIMA) and Long-Short Term Memory (LSTM). Table 1 describes each predictor in the predictor pool of DF.

Figure 5: Predictor pool with 8 ML models and one LSTM model

Table 1: Predictors in predictor pool of DF

Predictors	Description
LR	• LR stands for Logistic Regression. • LR model use local history of WL to forecast the rate of job arrival. • This model is a single variable linear model as it considers only previous job arrival rate as variable. • The sample is selected using KNN function • This model can provide high accuracy for random WL pattern [1].
LASSO	• LASSO stands for Least Absolute Shrinkage and Selection Operator • Lasso regression is a type of regularization which is preferred for extra precise prediction. • This model makes use of shrinkage. That means, the data values are shrunk towards a central point known as the mean. This process is termed as shrinkage. • The LASSO technique encourages simple, sparse models (i.e., models with

	fewer parameters). • LASSO[2] is mathematically expressed as [**Residual Sum of Squares + λ * (Sum of the absolute value of the magnitude of coefficients)]**
RIDGE	• Ridge regression is a model tuning method used to evaluate data with multicollinearity. • L2 regularization is performed by this technique. • The accuracy varies, and shows variation from predicted values when there is an issue with multicollinearity, least-squares are unbiased, and variances are large[2]. • Cost function of RIDGE is expressed as **Min (||Y – X(theta)||^2 + λ||theta||^2)**
SVR	• Support Vector Regression refers to Supervised Machine Learning Models and associated learning algorithms that evaluate data for classification and regression analysis. • SVR is built on the Support Vector Machine (SVM) concept, which widely used for classification problems when the data is not linearly separable. • This model works well for both overall and specific workloads[2].
SGD	• Stochastic Gradient Descent (SGD) is a straightforward but highly efficient technique for fitting linear classifiers and regressors to convex loss functions such as (linear) Support Vector Machines and Logistic Regression. • Though SGD is a faster predictor, its convergence path is noisier than that of original gradient descent. This is due to the fact that the gradient is only approximated in each step. As a result, we see a lot of fluctuations. However, it is a far superior option[2].
LAR	• Least Angle Regression (LARS) is an alternative, efficient method of fitting a Lasso regularized regression model that does not require any hyperparameters. • The LARS model can be used to fit high-dimensional problems more efficiently[2].
HUBER	• The Huber Regressor optimizes the samples' squared loss and absolute loss, that is **{ | (y - X'w) / sigma | < epsilon }** and **{ | (y - X'w) / sigma | > epsilon }** respectively, where w and sigma are parameters to be optimized. • The parameter sigma makes sure that if y is scaled up or down by a certain factor, epsilon does not have to be rescaled to maintain the same robustness. • It should be noted that this does not account for the fact that X's various features may be of varying scales. • This ensures that the loss function is not significantly influenced by outliers while also not completely ignoring its impact[2].
ARIMA	• Autoregressive Integrated Moving Average, it is an abstraction of ARMA • By integrating AR and MA models, ARIMA offers a trustful prediction of non-stationary time-series data.

[2] https://www.mygreatlearning.com/blog/

	• ARIMA is expressed as ARIMA (p, d, q), p : AR's order q: MA's order d: differencing model's order • ARIMA have greater accuracy for the cyclic bursty WL that has strong trend and cyclic changes [1].
LSTM	• Long Short Term Memory model is the part of deep learning. • It consists of an array of memory to store the past data. • This model uses Adam's optimization technique. Adaptive Moment Estimation is a technique for optimizing gradient descent algorithms. • Adam optimization is used because the system that dealing with have broad amount of data or parameters. Adam's optimization technique is extremely efficient for such scenarios. Also, it utilizes less memory and is more efficient.

4.2 Workload Repository

The forecast record of each local predictors in the predictor pool is saved in the workload repository. This history is used for next prediction and well as the creation of ensemble model. From this repository, the MAPE value is compared and final predictor is selected for forecasting. The results are stored in repository as normalized form.

4.3 Ensemble model builder

DF create models for each predictor. Therefore, ensemble model builder is another main component of DF. The figure 6 depicts the steps of ensemble model builder. Train/test dataset, normalization, evaluation and creation are the steps involved in ensemble model builder. In order to train the model, the 20% of data in the dataset is used for testing purpose and rest for the prediction. Training of the predictor is necessary for the accurate prediction.

As the job arrival rates varies dynamically with time, normalization plays an important role to process the data in predictors. So, the pre-processed dataset is normalized using MinMaxScaler and StandardScaler functions. MinMaxScaler scale each feature to a range to transform. e.g., between -1 and 1. The mean of the observed values is zero and the standard deviation is one because the function called StandardScaler is used to resize the value distribution.

Evaluation of predictor is carried out using MAPE value. The value of MAPE for the models created by each predictor is compared and form a set of four predictors with less MAPE value. Then the ensemble model is created for these predictors and appropriate predictor has chosen from the set. Ensemble model creation interval is one second (by observing the result generation).

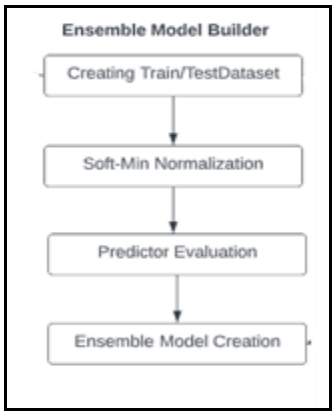

Figure 6: Ensemble model builder of DF

5 Implementation

DF is implemented using python 3.10.0 version in visual studio code and command prompt with operating system Windows 10. To implement predictors in predictor pool, machine learning libraries and packages are necessary. Libraries needed are NumPy, Statsmodels, Pandas, pickle, scikit-learn and so on. The software Anaconda is also installed for getting the packages for machine learning and deep learning.

To achieve the goal of improving accuracy of forecast, it is preferable for local predictors to have deterministic processing times. This requirement exists since DF works using a resource manager who must appropriately prepare cloud resources prior to the actual job arrives. We use a grid search [1] to decide the parameters for the local predictors, with a tradeoff between precision and forecast overhead. Here the soft margin and kernel parameters in SVMs with values ranging from $10e^{-3}$ to $10e^{3}$ are considered. Real workloads are not used in parameter selection to ensure fair evaluation and avoid over-fitting. But, real workloads [11] are only used to assess DF's performance.

6 Evaluation

The amount of job arriving in a system varies dynamically with time. So, the WL pattern varies accordingly such as cyclic bursty, random, on and off, increasing and so on. Here we use dataset for the WLs from High Performance Computing (HPC) System [11]. The HPC WLs exhibits wide range of characteristics. The Grid 5000 WLs are bursty and random, whereas LCG and NorduGrid WLs are seasonal. That is ON and OFF and cyclic. These WLs contain 62.5K jobs, 435K jobs and 122K jobs respectively. Figure 7 illustrates the WL of pattern of these datasets.

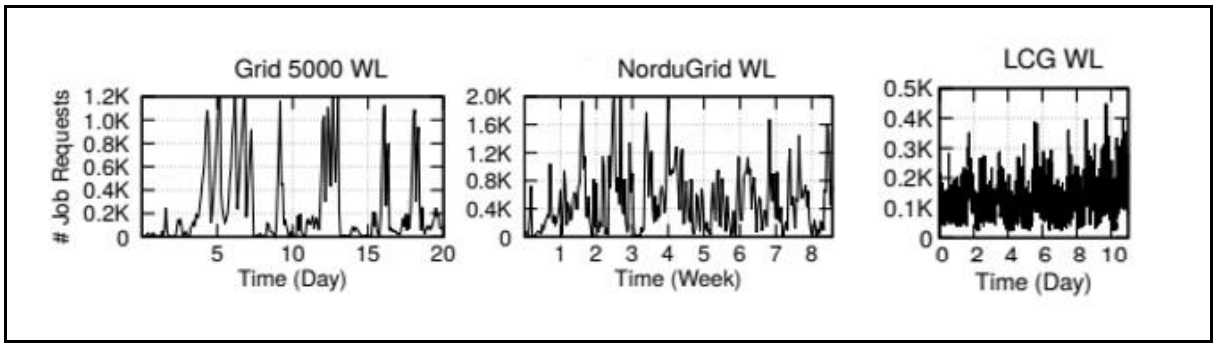

Figure 7: Workload pattern of Grid 5000, NorduGrid and LCG

Mean Absolute Percentage Error (MAPE) is measured, in order to evaluate the precision of DF . When MAPE decreases, the accuracy of prediction/forecast increases.

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \quad (1)$$

A_t - Actual Value

F_t - Forecast Value

DF is compared against another framework called CloudInsight and the predictor ARIMA. CloudInsight has chosen because of its “one fits all” approach. And ARIMA is selected as it is widely used in many predictive approaches.

The figure 8 illustrates the MAPE for the WL NorduGrid. It is evident that 9.6% is the MAPE obtained for DF with load having seasonal variation (on and off pattern).

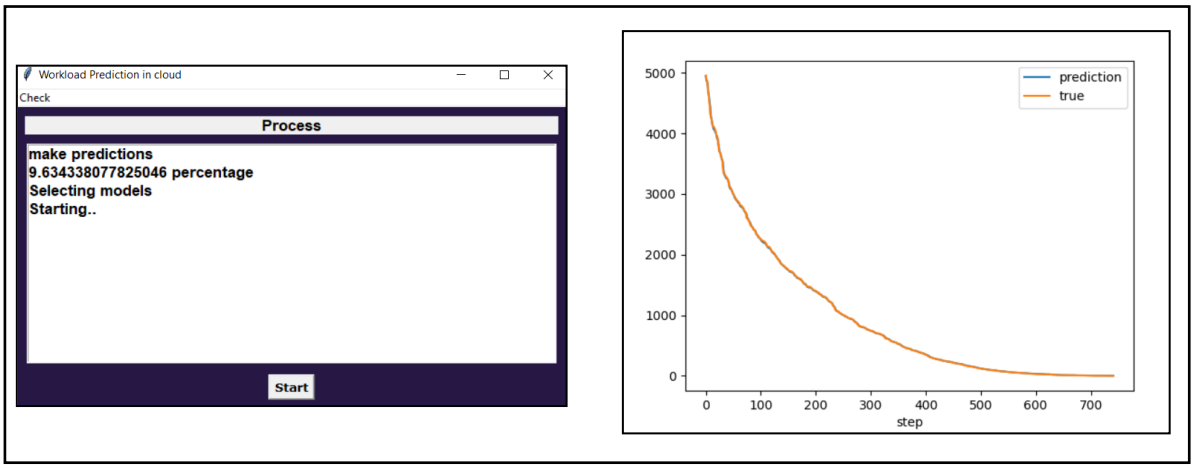

Figure 8: MAPE for DF with dataset NorduGrid among HPC loads with seasonal variation in Workload patterns

The figure 9 demonstrates the MAPE for the WL Grid5000. It is evident that 3.96% is the MAPE obtained for DF with load pattern bursty and random.

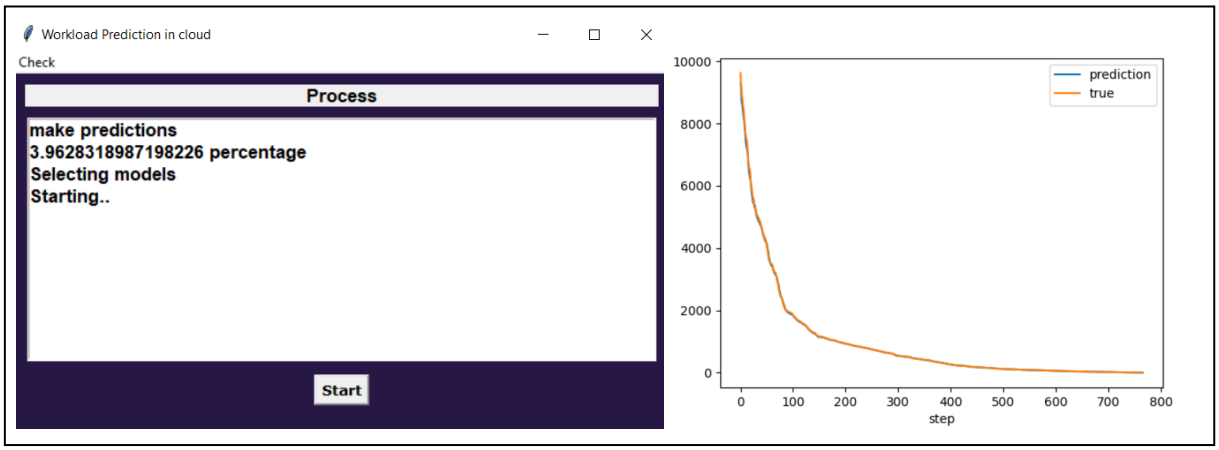

Figure 9: MAPE for DF with dataset Grid5000 among HPC loads with bursty and random Workload patterns

The figure 10 depicts the MAPE for the WL LCG. It is evident that 8.9 % is the MAPE obtained for DF with load pattern on and off or increasing.

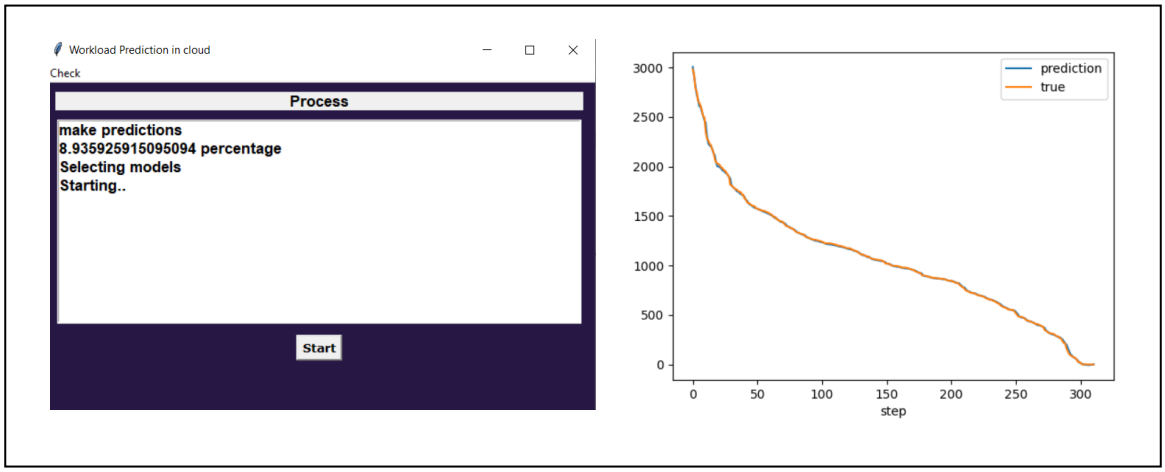

Figure 10: MAPE for DF with dataset LCG among HPC loads with seasonal variation in Workload patterns

While evaluating the obtained result with the CloudInsight and the predictor ARIMA, DF has less MAPE values for the dynamic fluctuating WL pattern. The table 2 illustrates the same.

Table 2: MAPE results with HPC workloads

Workload	**MAPE for DynamicForecast**	**MAPE for CloudInsight**	**MAPE for ARIMA**
Grid 5000	4%	54%	62%
NorduGrid	10%	20%	28%
LGC	9%	32%	38%

Therefore, it is clear that , DF has 93%, 50% and 72% of reduction in MAPE from CloudInsight and 94%, 64% and 76% of reduction in MAPE from the predictor ARIMA. As a result, the accuracy of DF is greater than CloudInsight and other prediction techniques. Figure 11 illustrates the comparison of DF with CloudInsight and ARIMA.

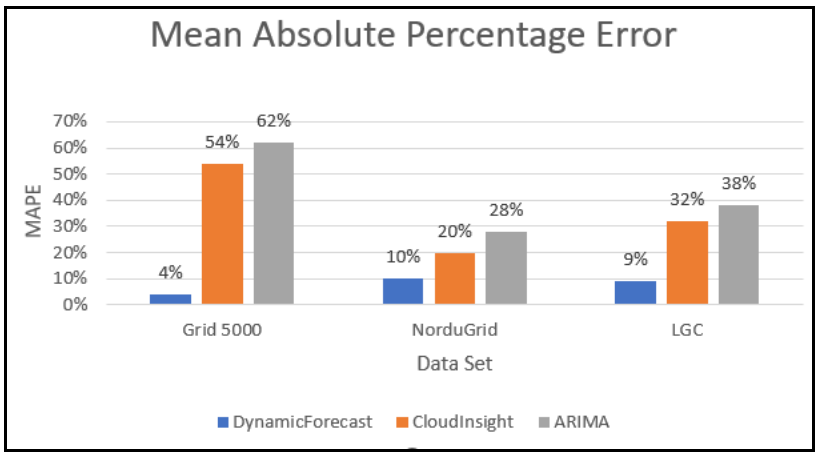

Figure 11: MAPE comparison for DF, CloudInsight and ARIMA

7 Conclusion and Future Work

This paper presents DynamicForecast- A council of workload Prediction framework in order to address the issues with dynamic and highly variable workload pattern and thereby its accuracy of prediction and management of resources in cloud. DF employs a stack of machine learning predictor models with Long Short-Term Memory Model along with Adam's optimization technique to build four ensemble models and carry out prediction more precisely. Results obtained indicates that, DF has 50% - 94% lower MAPE value in comparison with other current state of the art WL predictors (CloudInsight and ARIMA) for varying load patterns (ON and OFF, increasing, bursty and random) of HPC system.

Even though, DF shows lower MAPE than all existing predictors, we can expect an improved results with implementation of DF with 100% deep learning or AI technique.

References

[1] Y. Q. I. K. Kim, W. Wang and M. Humphrey, "Forecasting cloud application workloads with cloud- insight for predictive resource management," IEEE Trans. Cloud Comput., pp. 1–16, May 2020, doi:10.1109/TCC.2020.2998017. JCR Impact Factor 2021: 5.938.

[2] I. K. K. V. K. Jayakumar, J. Lee and W. Wan, "A self-optimized generic workload prediction frame- work for cloud computing," in IEEE Int.Parallel and Distrib. Process. Symp. (IPDPS), New Orleans, USA, May 1818-22, pp. 779–788, IEEE, 2020, doi:10.1109/IPDPS47924.2020.00085. CORE2021 Rank=A.

[3] H. W. J. Gao and H. Shen, “Machine learning based workload prediction in cloud computing,” in 29th IEEE Int. Conf. on Comput. Commun. and Netw. (ICCCN), Honolulu, HI, USA, Aug. 3-6, 2020, pp. 1–9, doi:10.1109/ICCCN49398.2020.9209730. CORE2021 Rank=B.

[4] Y. Q. I.K. Kim, W.Wang and M.Humphrey, “Cloudinsight: Utilizing a council of experts to predict future cloud application workloads,” in 11th IEEE Int. Conf. on Cloud Comput. (CLOUD), San Fran- cisco, CA, USA, July 2-7, 2018, pp. 41–48, doi:10.1109/CLOUD.2018.00013. CORE2021 Rank=B.

[5] J. Kumar, A. K. Singh, and R. Buyya, “Self-directed learning-based workload forecasting model for cloud resource management,” Inf. Sci., vol. 543, no. 1, pp. 345–366, Jan 2021, doi: 10.1016/j.ins.2020.07.012. JCR Impact Factor 2021: 6.795.

[6] Z. Gong, X. Gu and J. Wilkes, "PRESS: PRedictive Elastic ReSource Scaling for cloud systems," 2010 Int. Conf. on Netw. and Service Management (CNSM), Niagara Falls, ON, Canada, 2010, Jan. pp. 9-16, 2011, doi: 10.1109/CNSM.2010.5691343. CORE2021 Rank=B.

[7] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, “CloudScale: elastic resource scaling for multi-tenant cloud systems”, in 2nd ACM Symposium on Cloud Computing (SOCC '11). Association for Computing Machinery, New York, NY, USA, 2011 Oct. Article 5, pp 1–14. DOI: https://doi.org/10.1145/2038916.203892. JCR Impact Factor 2021: 4.79.

[8]US Department of Commerce; Available on: https://www.nist.gov/system/files/documents/2017/05/31/evaluation_of_cloud_computing_services_based_on_nist_800-145_20170427clean.pdf

[9] Y. Jiang, C. -s. Perng, T. Li and R. Chang, "ASAP: A Self-Adaptive Prediction System for Instant Cloud Resource Demand Provisioning," 2011 IEEE 11th International Conference on Data Mining, Vancouver, BC, Canada, 2012, Jan, pp. 1104-1109, doi: 10.1109/ICDM.2011.25. CORE2021 Rank=A*.

[10] J. Loff and J. Garcia, "Vadara: Predictive Elasticity for Cloud Applications," 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2015 Feb, pp. 541-546, doi: 10.1109/CloudCom.2014.161. CORE2021 Rank=C.

[11] Dataset; TU Delft. The Grid Workloads Archive available online on " http://gwa.ewi.tudelft.nl/datasets/?msclkid=d9112cc7b2ea11eca7067512037bc5d6"

[12] Nikolas Roman Herbst, Nikolaus Huber, Samuel Kounev, and Erich Amrehn, "Self-adaptive workload classification and forecasting for proactive resource provisioning," In Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE '13). Association for Computing Machinery, New York, NY, USA, 2013, 187–198. DOI:https://doi.org/10.1145/2479871.2479899. CORE2021 Rank=B.

[13] Chunhong Liu, Chuanchang Liu, Yanlei Shang, Shiping Chen, Bo Cheng, Junliang Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," Journal of Network and Computer Applications, Volume 80, 2017, pp 35-44, doi: https://doi.org/10.1016/j.jnca.2016.12.017. JCR Impact Factor 2021 : 8.17.

[14] A. Y. Nikravesh, S. A. Ajila and C. -H. Lung, "Towards an Autonomic Auto-scaling Prediction System for Cloud Resource Provisioning," 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2015, Aug, pp. 35-45, doi: 10.1109/SEAMS.2015.22. CORE2021 Rank=A.

[15] T. Lorido-Botran, J. Miguel-Alonso, J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," Journal of Grid Computing, December 2014, 12:4, pp 559–592, doi: https://doi.org/10.1007/s10723-014-9314-7. JCR Impact Factor 2021 : 3.986.

[16] R. N. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," in IEEE Transactions on Cloud Computing, 2015, Oct, 3(4), pp. 449-458, doi: 10.1109/TCC.2014.2350475. JCR Impact Factor 2021: 5.938.