MICRAST: Micro-Forecasting Approach for Cloud User Consumption Pattern Based on RNN

Shallaw Mohammed Ali, Gabor Kecskemeti
Institute of Information Technology, University of Miskolc, Miskolc, Hungary

***Abstract*—One vital key for effective management of cloud resources is the ability to predict their users' consumption's patterns in granular level. It can provide more insightful analysis to guide these users towards more resource-effective habits. Such prediction requires pre-processing the users' traces from these cloud resources for granular prediction (micro-prediction). However, the methodology followed by many forecasting based cloud studies was designed to deal with these traces as overall trends (macro-prediction). We propose a (MICRAST) that generates segments of granular patterns. Then, it carries out parallel tasks of pre-processing and training that lead to separate trained network for each of these segments. To select a model for our approach, we compared methods from two forecasting categories: statistical and artificial neural network (ANN)-based. The results lead us to recurrent neural networks (RNN). We evaluated the MICRAST through a comparison with related work methodologies (macro-prediction approach) for both uni-variate and multi-variate forecasting. Then, we measured its confidence for forecasting up to 20% of the training time steps. The results showed that our approach can forecast the preferences of each cloud user with a confidence level of between (95% to 98%) surpassing related works by more than 70%.**

***Keywords*—Micro-forecasting; cloud workload; data processing; macro-forecasting; data mining**

I. Introduction

The ability to forecast users' consumption preferences for any service provider can profoundly influence its resource management. Such ability has a high impact on shaping decision-making processes. Anticipating these preferences enables proactive decisions that align with users' requests. Effective forecasting ensures the identification of potential challenges and opportunities associated with resource utilisation. Furthermore, implementing forecasting into management frameworks can foster collaboration among diverse stakeholders. It's highlighted by [1] that accurate forecasting enables practitioners to respond efficiently to changing resource-related conditions. Similarly, researchers in [2] emphasised the role of users' behaviour and preference forecasting in enhancing the resilience of resource management solutions. They underscore its significance in achieving long-term sustainability goals.

Many studies, such as [3], [4], and [5], presented different types of forecasting models for similar purposes. For instance, in [3], researchers proposed a multivariate deep learning model to forecast workloads in data centers. Also, for better resource management, Lu et al. [4] presented a novel back-propagation neural network algorithm to predict future cloud logs. However, the limitation of the approaches for the current cloud forecasting models is that they were designed to predict based on the overall trace. In another words, they lack in capturing and predicting cloud traces in detailed, granular levels. Unfortunately, analysing users' traces as a whole is not beneficial for predicting individual usage preferences. In their raw form, these traces do not readily reveal the meaningful trends in historical records necessary for predicting individual preferences. Consequently, employing these models is not suitable for consumption-steering purposes.

Therefore, we propose a new forecasting approach to address the aforementioned challenge. Our approach has three main pipelines: extraction (segmentation) via clustering, pre-processing, and forecasting. In the first pipeline, we extract segment of hidden, fine grained pattern from the input trace by filtering and clustering it. Each segment represents the historical trends for each pattern. According to our findings in [6], clustering demonstrated the ability to perform such extraction with high accuracy. To ensure efficient clustering, this extraction involves using our recent technique of dimensions and method selection EFection [7] and SeQual [8]. Then, in the pre-processing pipeline, the segmented patterns are uniformed with time alignment and linear interpolation. Finally, in the last pipeline, the pre-processed data is used for training and forecasting for the prediction process. To select a suitable model for our approach, we conducted a preliminary evaluation experiment. In this experiment, we compare the performance of various statistical and ANN-based models. We choose a recurrent neural network (RNN) for our approach. Accordingly, we present this approach as a Micro-forecasting approach for cloud user consumption pattern based on RNN (MICRAST) .

We evaluated our approach through two experiments. First, we compared MICRAST performance with a sample forecasting model. We employed these two approaches to forecast each user's preferences from all indicated cloud traces. Then, we measured the prediction accuracy of the results against their actual preferences. To ensure accurate validation for various scenarios, we applied this evaluation to both univariate and multivariate forecasting. In the second experiment, we assessed the confidence of our approach over a range of prediction time steps. This was achieved by measuring the change in accuracy when gradually increasing the forecasting time steps up to 20% of the training data. To measure the forecasting accuracy, we used the coefficient of determination R^2 and the mean absolute percentage error (MAPE). Our approach demonstrated the ability to forecast user behaviour with an accuracy between 95% to 98% R^2 surpassing related works methodology by 70 percentage points.

We structured the rest of this paper as follows: In Section II, we cover the background of this study. This includes giving a brief description of the common forecasting models and the accuracy measures used to evaluate them. This is followed

by a presentation of the related works. Then, in Section III, we disclose the process for developing the MICRAST approach and the inquiry works that contributed to it. Next, in Section IV, we demonstrate the evaluation process for the proposed approach and the experimental findings. This includes a comparison between our approach and a case study that presents an example of the related work approach. Finally, Section summarises the main points of this paper and reveals our future plans.

II. Background

This section covers the essentials of forecasting in cloud computing. This includes presenting commonly used models and describing their validation metrics. Then, it introduces the typical cloud workload traces and their characteristics in terms of forecasting implementation. Finally, this section discusses the literature review for the related works.

A. Time Series Forecasting and its Models

In time series forecasting, prediction is performed based on data comprising a sequence of observations over time [9]. Two vital parameters of this prediction are the forecasting *window size* and *steps*. In this context, the window size represents the range of past events, a line of records in the trace, that are utilised to be captured by the forecasting models. While the number of future records to be predicted by these models is denoted by steps.

Forecasting models are typically categorised into two main types: statistical and ANN-based models. Statistical models, as the name indicates, use statistical techniques and assumptions about the data distributions to reveal trends in historical records for predicting future variables. While ANN-based models perform the prediction using artificial neural networks to analyse and learn from past records. In this section, we aim to cover the simplest to the advanced models of these two categories. These were selected with respect to their range of usability.

Accordingly, Table I presents these models with their category and uses. We started with one of the very basic statistical forecasting methods, the Simple Moving Average (SMA) [10]. It estimates the future data values by finding the mean of data collection points falling within a certain forecasting window [11]. SMA is best for short-term prediction of stable trend time series data. In the context of time forecasting, stability or stationarity means that its statistical properties (mean, variance, and auto-correlation) do not change over time. Another model is the Auto-Regression model (AR), in which the forecasting is performed through a linear combination of its past values. The AR model is flexible for different types of time series patterns [12]. To form an Auto-regressive Moving Average (ARMA) model, AR is combined with another type of MA, which uses past errors to predict the future event in a regression-like model [12]. In ARMA, the AR part predicts the current event based on the past one, while the MA part calculates the errors of past predictions to correct the current one. ARMA is suitable for a stable series with no trend or seasonality. From this mix, Auto-regressive Integrated Moving Average (ARIMA) was introduced by Box and Jenkins by adding integrated differentiating to ARMA for converting the targeted data to stability [13]. This makes ARIMA usable for non-stable time series as well as for both short-and long-term forecasting. However, it cannot detect non-linear characteristics in the data, such as abrupt changes or variable interactions [12].

It's important to note that the above-described models are applicable only for uni-attribute forecasting, as depicted in Table I. Thus, a Vector Auto-Regression (VAR) model was presented as the multi-attribute version of the statistical model that is used for multiple attribute predictions. In VAR, the next value for each attribute is predicted based on its own previous history in addition to the history of other related attributes [14]. In the context of cloud traces, the related attributes are those that represent the consumption records for the users in the same trace.

On the other hand, from the ANN-based forecasting models, this section covers the following: RNN, LSTM and GRU consist of closed loops of network connections and feedback. These networks are developed to learn a sequential pattern of time series data [15]. Recurrent Neural Network (RNN) is useful for stable time series data. However, according to Bengio et al. [16], one of the limitations of RNN is the challenge of vanishing gradients when the forecasting window increases. These gradients used to update the network's weights during training. This makes the network struggles to learn from earlier time steps, making it hard to capture long-term dependencies in the trace.

To overcome this challenge, the literature introduced the concept of Long Short-Term Memory (LSTM). LSTM accomplishes this overcoming by discarding irrelevant information in the network using gating mechanisms, which enable them to deal with long-term forecasting windows [17]. Cho et al. [18] proposed an improved version of RNN with gate optimisation called Gated Recurrent Unit (GRU). GRU has a similar structure to that of LSTM and is also used to address the issue of vanishing gradients in time series forecasting. It is worth mentioning that an essential advantage of ANN-based models is that they can be employed for both multi-attribute and uni-attribute forecasting scenarios. Table I presents these models and their uses.

TABLE I. The Discussed Forecasting Models

Model	For	Category
SMA AR ARMA ARIMA	Uni-attribute	Statistical
VAR	Multi-attributes	
LSTM GRU RNN	Both	ANN-based

B. Data Analysis and Selection

In the forecasting area, the majority of prediction models are based on the assumption that the data of interest is stable [19]. Such stability indicates that the statistical properties of this data do not change through time, which makes it simpler to analyse the prediction process. Accordingly, our cloud traces need to be analysed for stability to ensure efficient forecasting. Without meeting the stability condition, the forecasting results may turn out to be unreliable. Typically, to check the stability

of the targeted traces, unit root tests are used. And to perform the unit root test, several types of methods are employed. Among others, these methods are Augmented Dickey-Fuller (ADF), Phillips-Perron (PP), and Zivot-Andrews [20]. According to [21], one of the most commonly used methods is ADF. It tests the data according to the following two hypotheses:

- Null hypothesis: The dataset has a unit root, and thus it's non-stable.
- Alternate hypothesis: The dataset doesn't have a unit root, and it's stable.

Therefore, we checked the stability of the cloud traces from the resources of the grid workload archive and the parallel workload archive to ensure efficient forecasting. To this end, we employed the ADF test for its high efficiency, being the most commonly used test in the related literature. We applied this test to users' preferences of (Requested Number of Processors) for all the traces in Table V, as it reflects their consumption records. Then, we calculated the average of ADF's results for the corresponding trace.

The results showed a P-value below 0.05, which represents the threshold of stability for all these traces with ADF statistic values shown in Table II. These results ranged from (-3.5) to (-20) for all the supervised traces (and only Bitbrain in the unsupervised trace). This means that all the traces from the selected resources are below the standard critical values that are used in the literature; see Table II. And since the P-values for each cloud trace were below 0.05, the null hypothesis is rejected, and these traces seem to be stable. Nevertheless, the strength of stability is not on the same level for all these traces. The farther the traces statistic is from the critical value, the stronger its stability. For instance, the CTC SP2 with (-20) can be considered to have very strong stability. While the UNILU Gaia, which recorded the lowest statistic value of (-3.5), has the lowest stability from these traces and requires more careful processing in the forecasting.

TABLE II. ADF CRITICAL VALUES

Level of Significance	Critical Value
1%	-3.43
5%	-2.862
10%	-2.567

Despite exhibiting high stability, many of these traces, such as ANL-Intrepid, SDSC Par 1995, OSC Cluster, and CEA Curie, showed non-linearity with abrupt changes. They also recorded a high standard deviation (SD) of above 10K. This is noticed when their scales are examined, such as the example provided in Fig. 1. We concluded that cloud workload traces may exhibit a characteristic of irregular changes without following a seasonality, yet still maintain a sense of stability. Such characteristics require a pre-processing to reveal meaningful patterns and trends from these traces to be beneficial for prediction model training.

Furthermore, since our analysis framework aims at providing a micro-prediction approach, it's vital to evaluate this approach with cloud traces that are applicable for such an aim. To be applicable, these traces need to meet the following criteria:

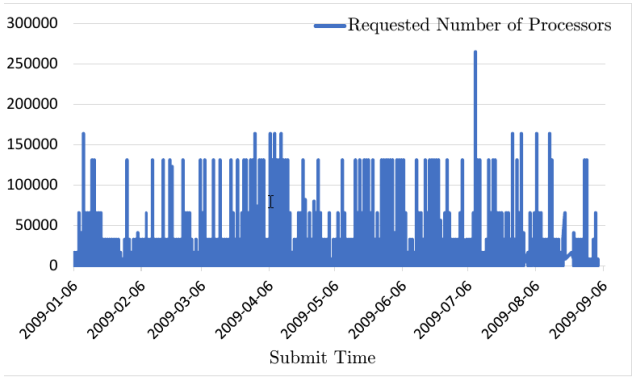


Fig. 1. The Characteristic of ANL-interpad trace.

- The trace should provide the attributes that present users' preferences in numerical format. Such a format makes it possible to extract these patterns and enables the forecasting model to capture them more efficiently.
- The trace should provide job submission times for each user. This enables forming a history of sequences of events for these users based on their job times. These sequences are essential to enabling the forecasting models to learn past preferences.
- The size of the trace should be sufficiently large to enable effective learning. In our experience, ANNs have a hard time learning trace time series with less than 25K data points, so we expect such trace size at least from each suitable for us to work with.
- The trace should demonstrate a sense of stability, since most forecasting models assume that the characteristics of the targeted datasets are stable.

Based on the above, we selected the traces in Table V that meet the above criteria. This table represents the trace name, its size (in number of lines), the time period of the trace, and the ADF test results.

It's vital to emphasise that another prerequisite for data to achieve efficient forecasting is that it should be uniformly sampled. This is necessary when the information in this data is given on different scales. But what to do when we don't have the dataset collected in a uniformly sampled way? Such uniformity can be achieved with the implementation of time alignment and linear interpolation methods. Time alignment ensures that the action points in the data are synchronised to the corresponding time step they represent [22]. Linear interpolation, on the other hand, fills in the blanks where there is no data. In essence, it is joining two known values with a straight line and then carrying out approximations for the intervening ones [23]. We provide two samples of time series data from the ANL-interpad trace. Table III shows the trace structure before applying the uniforming process, which shows obvious unaligned time steps or users' IDs. While Table IV shows how the same trace changed to a more uniform characteristic after applying time alignment and linear interpolation methods.

1) Forecasting validation: Forecasting validation is the process of measuring the efficiency of the employed model to

TABLE III. TIME SERIES SAMPLE OF ANL-INTERPAD TRACE BEFORE UNIFORMING (TIME ALIGNMENT AND LINEAR INTERPOLATION)

Submit Time	Requested Number of processors	User ID
2009-01-01 00:00:00	2048	1
2009-01-01 00:00:07	2048	1
2009-01-01 00:26:30	2048	1
2009-01-01 00:36:45	8192	2
2009-01-01 00:42:46	2048	1
2009-01-01 00:45:51	64	3
2009-01-01 01:31:25	16384	4
2009-01-01 01:49:13	64	3
2009-01-01 02:52:35	64	3
2009-01-01 03:55:58	64	3
2009-01-01 03:58:33	2048	1
2009-01-01 06:05:41	2048	1
2009-01-01 07:22:26	2048	1
2009-01-01 07:38:41	2048	1

TABLE IV. TIME SERIES SAMPLE OF ANL-INTERPAD TRACE AFTER UNIFORMING (TIME ALIGNMENT AND LINEAR INTERPOLATION)

Submit Time	Requested Number of processors	User ID
2009-01-01 00:00:00	2048	1
2009-01-01 01:00:00	2046	1
2009-01-01 02:00:00	2044	1
2009-01-01 03:00:00	2042	1
2009-01-01 04:00:00	2040	1
2009-01-01 05:00:00	2038	1
2009-01-01 06:00:00	2036	1
2009-01-01 07:00:00	2034	1
2009-01-01 08:00:00	2032	1
2009-01-01 09:00:00	2030	1
2009-01-01 10:00:00	2028	2
2009-01-01 11:00:00	2026	2
2009-01-01 12:00:00	2024	2
2009-01-01 13:00:00	2022	2

predict future events. It is typically conducted by comparing the outcome of a prediction with the actual ground truth. In the context of cloud traces, not all the datasets are applicable for training and using a portion of it as ground truth, since they lack a sufficient amount of usable records for such a purpose. The forecasting validation is mainly implemented to check if the used model is accurate enough in the testing process of forecasting.

Four of the most well-known of these validation metrics are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) [24]. MAE is a statistical metric that evaluates the overall accuracy of a regression model by averaging the absolute differences between predicted and actual values. In contrast, MAPE calculates the average absolute percentage error, providing a relative measure of prediction accuracy [25]. It presents the results on percentage scales from 0 to + ∞ (where 0 is the best). This makes the MAPE metric easier to interpret. Thus, it was a widely used metric for forecasting evaluation [26]. While RMSE is a measure of how far off a model's predictions are from the actual values. Similar to MAE, RMSE presents the quality for the predicted value in units as actual numbers, without expressing its relativity to the true value.

On the other hand, R^2 is a statistical measure of the linear relationship degree between two data variables [27]. It ranks the relationships between the predicted and actual values. The R^2 ranges its results between 0 and 1, where closer to 1 means better.

Notably, both R^2 and MAPE provide a clear scale for measuring forecasting accuracy. These metrics can accurately measure the degree of alignment between actual and predicted data. They also demonstrate a clear and accurate comparison across different forecasting models. As we discussed above, the accuracy measures for these metrics are presented as percentage-based values. While, other metrics, such as RMSE use actual values that may not be comparable. Based on these points, we employed MAPE and R^2 for the evaluation process of this paper.

C. Related Works

In the area of cloud computing, researchers have developed various forecasting models for different purposes. Most of these models were especially aimed at addressing the challenges of dynamic resource management and scaling.

In [28], Lu et al. proposed a model called RVLBPNN to forecast workload trends based on their historical data. This was combined with workloads' level of latency sensitivity. Later [29] presented an improved version of RVLBPNN through exploiting the use of the K-means clustering method. This new version predicts future workload trends based on the history of response time characteristics for these workloads.

Maiyza et al. [30] also aimed to target and predict workload values and future trends through presenting VTGAN, a nonlinear prediction model. In [31], Arbat et. al. proposed a time-series forecasting model designed to predict changes in cloud workloads with high accuracy and low inference overhead. The model used in this paper is called WGAN-gp Transformer. This model is inspired by the Transformer network and improved Wasserstein-GANs. It aims to address the challenges of the dynamic nature of cloud workloads.

Kumar et al. [32] developed an LSTM/RNN-based model to enhance resource management and optimise performance by accurately predicting future workloads, which is crucial for efficient operation in cloud environments. It predicts workload values based on their previous samples. The authors also presented a similar forecasting approach in [33], embedding a self-directed learning process to predict future demand from cloud servers.

Likewise, in [5], a MAG-D model was developed by Zhang et al. based on a GRU neural network. This model predicts each cloud resource's memory and CPU usage based on data-centre traces. On the other hand, to forecast user behaviour trends in large-scale cloud environments, Panneerselvam et. al. [34] implemented the InOt-RePCoN model. The trends that this model aimed to predict were the number of jobs and submission times for users. Similarly, in [35], Nehra and Kesswani presented a LSTM-based forecasting model to predict workloads in a cloud computing environment. Its aim is to reduce service level agreement violations.

We have concluded from above that the models in these studies have followed a similar forecasting approach. The

TABLE V. CLOUD WORKLOAD TRACES SELECTED FOR FORECASTING INVESTIGATION

Trace	Trace size	Time period	ADF statistics
KTH-SP2-1996	≈30K	Sep-1996 to Aug-1997	-5.3
UNILU Gaia	≈50K	May-2014 to Aug-2014	-3.5
ANL-interpas	≈70K	Jan-2009 to Sep-2009	-13.7
SDSC-SP2-1998	≈75K	Apr-1998 to Apr-2000	-4.9
CTC-SP2-1996	≈8K	Jun-1996 to May-1997	-20.2
KIT-FH2-2016	≈100K	Jun-2016 to Jan-2018	-6
META CENTRUM- 2009		Dec-2008 to Jun-2009	-11.6
LLNL Thunder-2007		Jan-2007 to Jun-2007	-6.7
LANL-O2K		Nov-1999 to Apr-2000	-7.5
LANL CM5 1994	≈200K	Oct-1994 to Sep-1996	-19
HPC2N		Jul-2002 to Jan-2006	-12
RICC-2010	400K	May-2010 to Sep-2010	-12.11
CEA Cuire-2011	≈700K	Feb-2011 to Oct-2012	-10
PIK-IPLEX		Apr-2009 to Jul-2012	-5.5
SDSC-BLUE-2000	≈240K	Apr-2000 to Jan-2003	-17
LLNL-Atlas	≈50K	Nov-2006 to Jun-2007	-7
Sandia ross 2011	≈60K	Nov-2011 to Jan-2005	-6
OSC Cluster	≈80K	Apr-2000 to Nov-2001	-4.4
DAS2	≈200K	Jan-2003 to Jan-2004	-5.6
BitBrain	≈1M	Oct-2012 to Feb-2013	-7.9

forecasting process under these approaches targets and macro-predicts the overall values and trends of workloads. Such methodology is designed to deal with users' preferences in the traces as a whole. Unfortunately, these traces as a whole in their raw form do not reflect any meaningful patterns for prediction. Thus, the gaps in the current models is that they lack an efficient tool to uncover and capture the diversity and variability of users' consumption at a granular level.

III. METHODOLOGY

In this section, we cover the research that leads to our new forecasting approach. This approach aimed at providing more efficient micro-prediction of clouds granular patterns. MICRAST overcomes the challenging characteristic of the cloud users' records, which suffers from sudden changes in their requests as illustrated in Fig. 1. Such characteristics are not readily predictable by the models in the related works.

MICRAST offers an outline that enables micro-prediction through extracting segments of granular patterns from cloud traces using clustering and the efficiency tools of (SeQual [8] and EFection [7]). This was conducted upon proving that the cloud traces demonstrate a sense of stability as depicted in the analysis investigation in Section II-B, page 856, which aligns with the requirements of most forecasting models. This is followed by performing a comparison test between statistical and ANN-based forecasting models to select the best among them for our approach. We finalise this section by giving a thorough description of our proposed approach.

A. Forecasting Model Comparison

It's essential for developing an efficient forecasting approach to carefully select its model. Therefore, we carried out a comparison evaluation between various models listed in Table I to select the one that shows the highest performance in predicting cloud traces.

1) Setup configuration: We set up the number of input layers based on the formula (Number of attributes × window size). The window size represents the segments of the targeted traces that are selected for the forecasting model to capture in the learning process, as illustrated previously in Subsection II-B. In the hidden layer, the desired model is selected (either RNN, LSTM, or GRU) with 100 units. Choosing 100 units ensures a balance between the complexity of the model and computational efficiency. This number is sufficient to capture intricate patterns within the cloud traces attribute without causing overfitting or incurring high computational costs. Such configuration enables the network to capture the necessary patterns within the time series data.

This is followed by configuring the activation functions. These functions are essential elements in the neural network since they indicate the activation status of the correspondent neuron. Accordingly, we selected the (*tanh*) for learning and the (*Hard sigmoid*) for the recurrent layer, as prior art indicated that these functions typically result in higher performance [36]. Finally, the hidden layer is further connected with the third layer (the out layer). In this layer, the network is structured as one unit (output), and the ReLU activation function is selected to handle the output recurrent process. We present the implementation of the RNN network in the K-nime toolkit in Fig. 2 as an example of the above configuration.

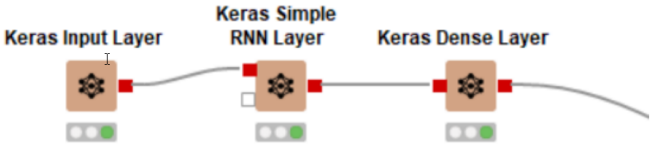


Fig. 2. RNN network configuration in K-nime.

2) Experimental implementation: As mentioned previously, we are developing our approach for uni-attribute and mult-attribute forecasting scenarios to ensure a wide range of ap-

plicability. Therefore, we performed this comparison through two experiments, one for each scenario.

We conducted experiments by comparing the performance of the forecasting models that were listed in Table V in the prediction of granular patterns. For this purpose, we chose the attributes of (Used Memory and Requested Number of Processors) as it represents the user's requests (consumption) from the cloud server provider. This attribute is widely available and applicable for forecasting across most traces. While others, such as Requested Memory, are deemed inapplicable since they exhibit a large portion of constant values in many traces, as we observed in [8] and [7], making it limited in giving meaningful information.

For the uni-attribute forecasting scenario, we targeted predicting the granular patterns in the Requested Number of Processors by training the model with its own historical records. While, for the multi-attributes scenario, we repeated this process by training the model on the historical records of an additional attribute (i.e. Run Time). We chose this attribute as we observed that it shows a high correlation with the Requested Number of Processors and it is also provided in applicable form in all the traces. This makes it a strong candidate for our purposes. It will test these models' ability to capture the dependencies between different attributes to predict a particular preference.

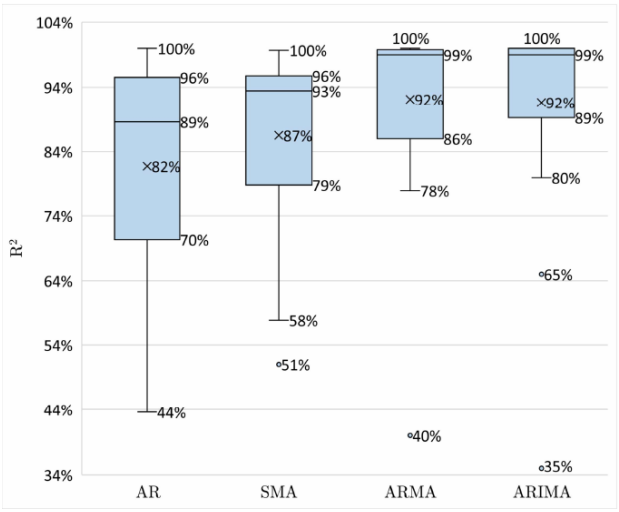
We applied both experimental scenarios to all the traces in Table V. At this time, we have prepared the input data by hand without presenting an automated approach. We clustered the targeted attributes to perform the extraction that allows the comparison to go ahead. We set the forecasting window size to five to ensure sufficient capture of past events. We observed that a window size of fewer than five might not sufficiently capture users' patterns in cloud traces.

In the ANN-based models, this setup is translated into configuring up to five input neurons and one output neuron with an activating sequence return. This results in five inputs for each chosen trace attribute's historical pattern. These configurations were applied to the uni-attribute scenario. While, for the multi-attribute scenario, the input neurons will be doubled to 10 to represent both attributes.

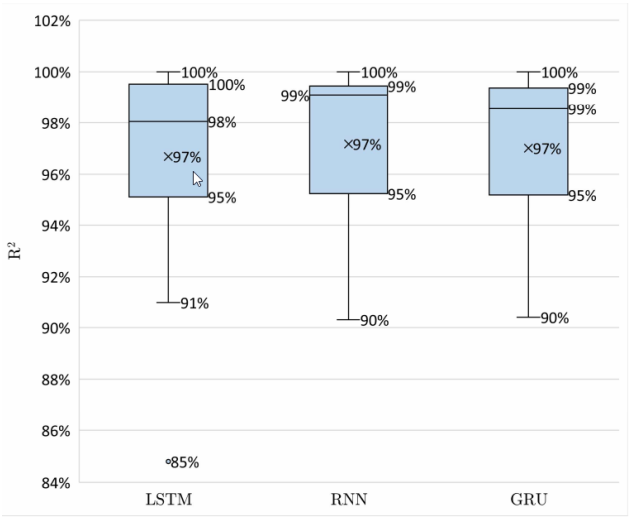
To implement these experiments, we split the prepared data into two portions: 70% for training and 30% for testing. To assess the accuracy of the forecasting, the metrics MAPE and R^2 were calculated to measure how closely the predicted values compare to the actual ones. Finally, we show boxplots for the distribution range of R^2 for these models to compare their performance. These boxplots provide insight into the precision and consistency of each model's performance. The results of our experiments are illustrated as follows:

a) Uni-attribute forecasting: In this experiment, we chosen the models that are useful for uni-attribute forecasting in Table I. Fig. 3 illustrates the boxplots of R^2 distribution ranges for the ANN-based and statistic models.

Fig. 3a showed that the basic statistical models (i.e. AR and SMA) exhibit a lower median and a wider range of R^2 distribution compared to the more advanced models (i.e. ARMA and ARIMA). We noticed this for the AR model with whisker extending down to 44%. This performance was mainly



(a) Statistical-based Models



(b) ANN-based Models

Fig. 3. R^2 Distribution for uni-attribute forecasting models.

caused by the traces of ANL-Intrepid and METACENTRUM-2009. This resulted in an average MAPE of around 2.1%. The SMA model exhibited a relatively higher median and distribution, with a slightly better whisker at 58% recorded for DAS2.

However, this model suffers from an outlier at 51% for HPC2N and a higher average MAPE of about 4%. On the other hand, both the ARMA and ARIMA models showed a comparably higher R^2 and narrower range of distribution. This implies that the more sophisticated models are more precisely focused and consistent than the basic models. Despite these elevated scores, both models suffered from a lower whisker of below 81% with outliers falling under 50% caused by the CEA Curie trace. This causes a higher average MAPE for these two models of around 12%. The reason for the performance of both simple and advanced statistical models is the nonlinear nature of users' consumption records in the above-indicated traces. Thus, they cannot be accurately

predicted with linear auto-regression and statistical analysis, even in advanced models. This underscores the uncertainty and unreliability of these models in forecasting users' preferences in the cloud environment.

In contrast, Fig. 3b illustrates that the ANN-based models show more stable performance in terms of R^2, with a higher median and a narrower range of interquartiles. All three models of LSTM, RNN, and GRU achieved the same high and concise range of R^2, except for an outlier at 85% for the LSTM model, which is the accuracy result of the LLNL ATLAS trace. This outlier is resulting in an average MAPE of around 0.7 compared to a 0.4 average MAPE for both RNN and GRU. In conclusion, we can assert that for uni-attribute forecasting, the ANN-based models (especially RNN and GRU) are more compatible with our approach to predicting cloud users' preferences.

b) Multi-attributes forecasting: In this experiment, we utilised the models that are applicable for multi-attribute forecasting in Table V. The boxplot in Fig. 4 shows the performance of both statistical and ANN-based models. As shown in Fig. 4, the VAR model exhibited a lower median, a wider range of R^2 distribution, and more outliers compared to the ANN-based models. Specifically, the outliers accounted for 72% in forecasting the DAS2 trace, 67% for SDSC BLUE, and 58% for CEA Curie. As mentioned previously, the attributes in these traces have the characteristic of non-linearity. Thus, these results demonstrate that the straightforward autoregression process of the VAR model cannot capture the correlation between attributes with such characteristics. It would rather interpret the patterns in these attributes as noise, resulting in an average MAPE of around 3.57%.

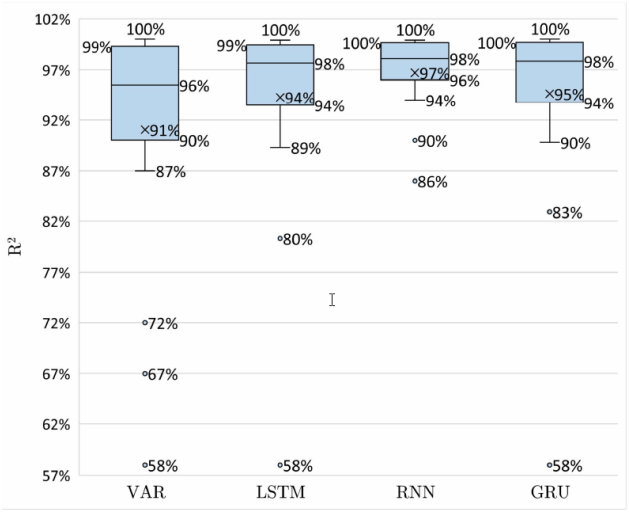


Fig. 4. An Accuracy distribution for multivariate forecasting models.

On the other hand, Fig. 4 shows that ANN-based models were able to handle such challenges with better performance and an average MAPE of around 1.9%. These models were able to capture the relationship between these attributes to predict the targeted preferences. However, both LSTM and GRU models suffered from an outlier at 58% R^2 for OSC Cluster. Notably, this trace recorded 95% in the uni-attribute forecasting for the same models. The reason for this drop is that using additional attributes (RunTime in this case) with (Requested Number of Processors) led to overfitting problems for both models. Such overfitting happened more for the DAS2, SDSC BLUE, and CEA Curie traces. The long-term memory for LSTM and GRU models causes such overfitting when trying to capture the relationship between two attributes with the significant characteristic of abrupt change. In comparison, the RNN model, with its more simple memory structure, showed the ability to deal with this, having more stable performance and achieving a narrower boxplot. In addition, compared to the previous ANN and statistical models, it achieved 96% accuracy in forecasting OSC Cluster, 96% for DAS2, 93% for SDSC BLUE, and 90% for CEA Curie. This implies that the RNN model effectively managed the noisy patterns and overfitting issues while maintaining high accuracy.

c) Findings: We concluded that, among the compared models, the RNN model achieved a high accuracy across both uni-attribute and multi-attribute forecasting. It recorded around 97% R^2. This model was able to maintain this performance even for challenging traces. This makes it an ideal choice for our approach. The detailed structure for MICRAST and its RNN network is detailed in the upcoming subsections.

B. The Proposed Approach MICRAST

We propose the MICRAST approach to predict the future consumption preferences of cloud users. Our approach achieves this through pipelines of segmentation, pre-processing and Forecasting as shown in Fig. 5. In this section we cover both the training and forecasting phases of our approach compared to the current approaches that shown in Fig. 6.

a) Training phase: this phase is carried out as following:

- Extraction pipeline is employing the use of clustering to uncover hidden patterns that steer users' preferences from the input trace. According to our findings in [6], clustering demonstrated a high ability of such extraction. To ensure efficient clustering, we apply two main tasks. First, we filter the trace by disregarding those attributes that have the same value for more than 80% of the records. These attributes are deemed unsuitable for clustering, as we illustrated in [8].

Second, we employ both tools of Sequential method of clustering Quality (SeQual) and Effectiveness detection of clustering quality (EFection), to address both scenarios of single and multiple feature selection. The SeQual method ranks which single attribute is best among the given for extraction when the user decides to process uni-attribute forecasting. While the EFection technique is used to select the combination of attributes that are more compatible for extraction when the user decides to process multi-attribute forecasting. Notably, if the EFection selected one attribute, in this case the user recommended going for uni-attribute forecasting instead. We also exploit the use of EFection to choose the most suitable method for clustering the selected attributes (for extraction). In this task, the selected clustering method groups similar historical usage records along with their submit time to form the consumption pattern for each user; see Table V. Thus, the output of this task are segments of granular patterns.

- Parallel pipelines of pre-processing to prepare each segment of granular pattern for prediction. In their clustered form, these segments exhibit non-uniform scales and formats. This form does not meet the requirements presented in Section II-B, see page 855, for effective data forecasting. Therefore, in these pipeline, we carry out uniforming processes in parallel, separately for each segment, as shown in Fig. 5. First, we take the current time sequence for each segment and convert them into a single form across all traces. We also employ the time alignment process to rearrange these segments on the same time scales. Second, we implement linear interpolation to avoid any missing records.

Afterword, the data of each segment are normalised into the range between 0 and 1. This is essential for efficient forecasting since the characteristics of cloud workload traces exhibit different scales of data. For instance, there is a significant difference in the standard deviation between Requested Time and Used Memory. Such characteristics are not suitable for forecasting, and normalising can make them more appropriate for an ANN forecasting model. The output of these pipelines are uniformed segments, each is ready to use as input for forecasting training.

- Parallel pipeline of forecasting that feeds the uniformed segments to train the RNN model. It's important to emphasise that the RNN model is configured with the setup presented in Section III-A1. This configuration demonstrated high performance, according to our comparative experiments. After training the RNN model sufficiently, this pipeline produce trained networks for each segment that will be ready for implementation to forecast the new input traces from the service provider system.

In addition, This pipeline involves also calculating the average centroid for each segment. These are stored alongside with each stored trained networks.

b) Prediction phase: In this phase our approach follows the same pipeline of segmentation and forecasting presented previously in the training phase. As shown in Fig. 6, first the new data are clustered into segments of granular patterns followed by calculating the average centroid for each of these segments. Finally these segments feeds into suitable trained networks to predict the future events. This is carried out by comparing the centroid of each segment with the one stored alongside the stored network from the training phase. Once the range of the compared average centroids matched, the correspondent network is selected and applied on the current segment for prediction.

In comparison to the approaches used by the recent cloud studies in Table VI, it's noticed that they follow a singular pipeline of prediction process. As illustrated in Fig. 6, these approaches are designed to carry out the data preparation (i.e. linear interpolation, time alignment, etc.) and forecasting tasks on the input data without considering granularity. As the prepared data feeds into the forecasting model to train a single network. While, for the prediction phase, this network applied directly on the new input data. Such prediction pipeline is known as Macro-prediction.

IV. Validation of MICRAST Performance

We conducted the evaluation in this work through two main experiments. First, we carried out a comparison test to measure the performance of our approach against (LSTM-RNN) in [32]. This case study exemplifies the micro-prediction approach that has been adopted by other related works as well in Table VI. We selected this study for the comparison evaluation since it is similar to our approach in aiming to predict consumption requests using an ANN-based model. Such evaluation is essential to demonstrate the benefit of one proposed approach. Second, we measured the forecasting confidence of MICRAST to show its performance across a scale of time steps. This is vital to show the application range for our approach. The following subsection presents these two experiments.

A. MICRAST vs LSTM-RNN for Related Work

In this evaluation, we compared the performance of MICRAST with the LSTM-RNN approach. We conducted this for both uni-attribute and multi-attribute forecasting scenarios to ensure accurate validation. To measure each approach's performance, we used R^2 and MAPE metrics. As illustrated in Section II-B1, we selected these metrics as they provide a clear scale for measuring forecasting accuracy. They measure the degree of alignment between actual and predicted data with a clear and accurate percentage-based value comparable across different forecasting models. We present the comparison results for each scenario.

Before we proceed to the results, we discuss the experimental configuration. Both forecasting scenarios adhered to the same evaluation setup described in the experimental implementation in Section IV using all the selected traces in Table V. Similarly, we first utilised both approaches to predict the consumption preferences for an attribute that represents users' usage records (i.e., Requested Number of Processors). Second, in the multi-attribute scenario, we repeated the previous steps with one difference: in this case, we train the forecasting models with the historical records of an additional attribute (i.e. RunTime). Accordingly, we are using the history of two attributes from the cloud trace to forecast the value of one particular attribute. We selected these attributes as they reflect the major aspects of consumption (demand level and duration).

1) Uni-attribute forecasting scenario: Table VII compares the average of R^2 and MAPE scores for forecasting all the selected traces by each approach. It demonstrates that our approach achieved better R^2 and MAPE by 67 and 40 per cent, respectively. These results showed a potentially significant improvement in accuracy when using our approach for uni-attribute forecasting.

For more detailed results, we presented the cumulative distribution for R^2 scores of both approaches in Fig. 11. Accordingly, the cumulative distribution for the LSTM-RNN approach in Fig. 7a showed below 60% R^2 for 16 out of 18 of the traces. In contrast, Fig. 7b showed that MICRAST recorded more than 90% R^2 for 17 of these traces.

We also demonstrated the MAPE for each trace in Fig. 8. The related work approach showed a significant MAPE for some traces. Specifically, it recorded around 165% to 209%

TABLE VI. MICRAST COMPARED TO RECENT APPROACHES

Approach	Prediction Focus	Granularity	Methodology	Prediction Level	Prediction Type
CNN-LSTM Model [37]	Multivariate cloud workload prediction	Medium (system-level)	Combines CNN for spatial features and LSTM for temporal dependencies	**Macro-prediction**	System-level workload forecasting
esDNN [38]	Cloud workload prediction & resource optimization	Medium (system-level)	GRU-based deep learning for time series forecasting	**Macro-prediction**	System-level workload & resource management
Facebook Prophet [39]	VM workload behavior prediction	Medium (workload-level)	Prophet framework with hyperparameter tuning and data preprocessing	**Macro-prediction**	Workload pattern forecasting (steady, trending, seasonal, etc.)
MICRAST	Individual user consumption prediction	High (user-level)	Pre-processing steps (clustering, uniforming, time alignment) + LSTM-RNN	**Micro-prediction**	Granular, personalized predictions

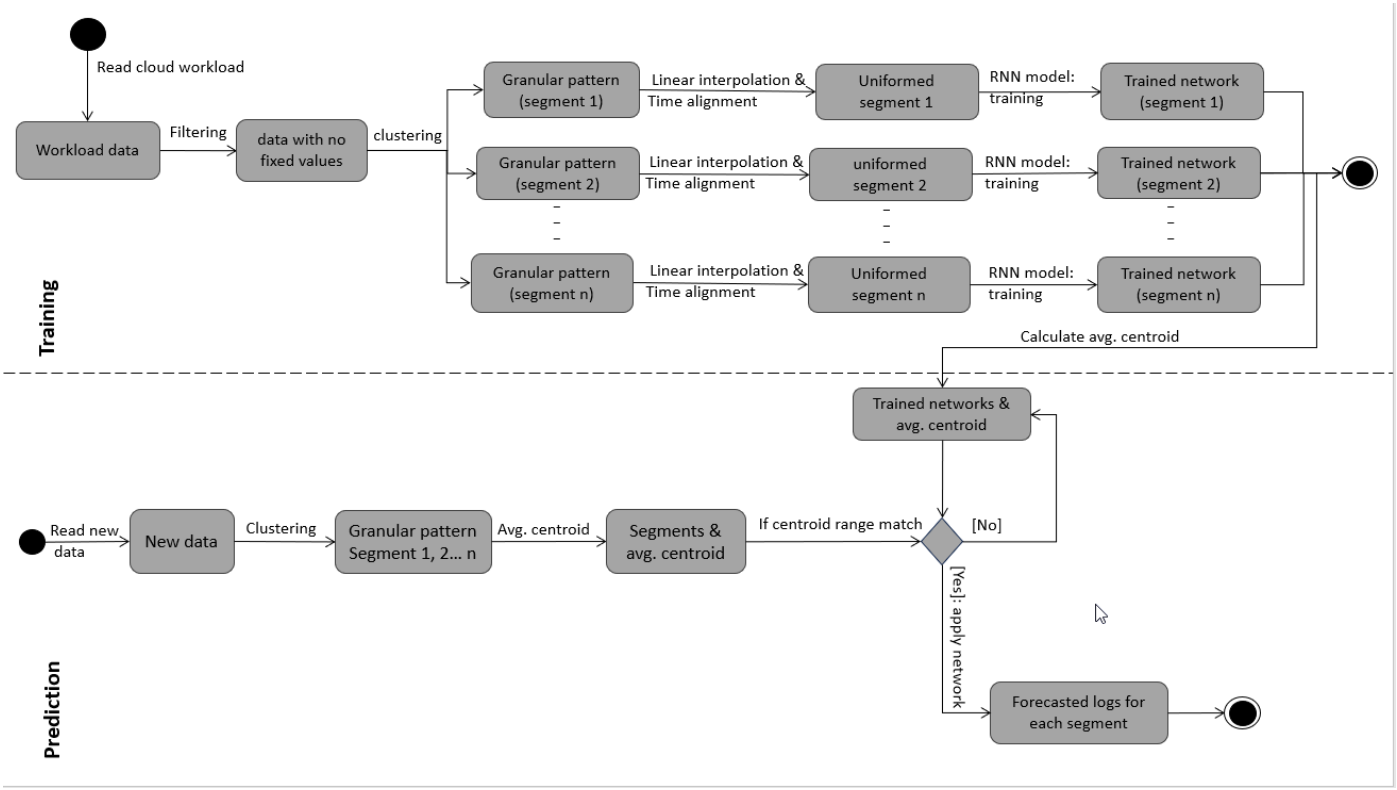


Fig. 5. The MICRAST approach.

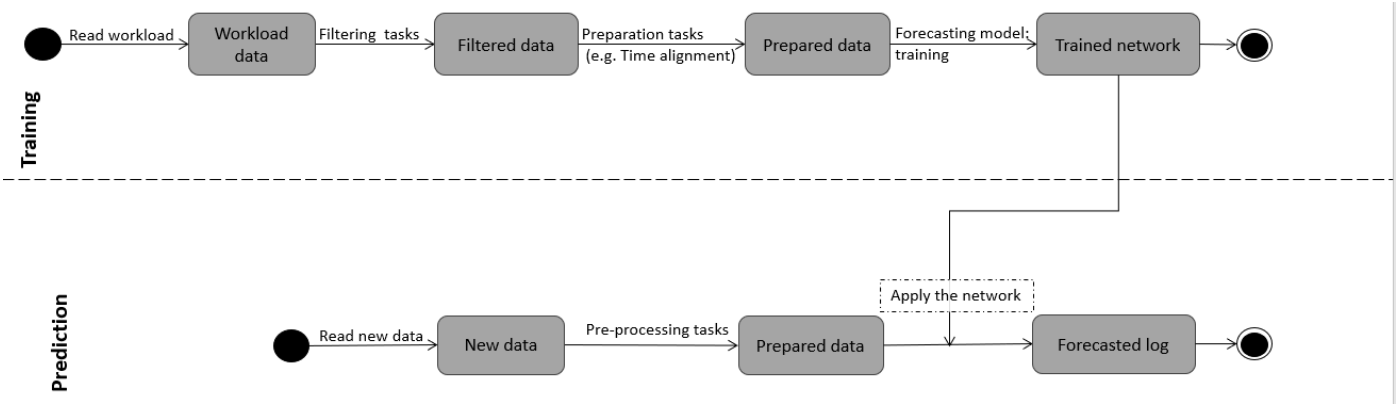
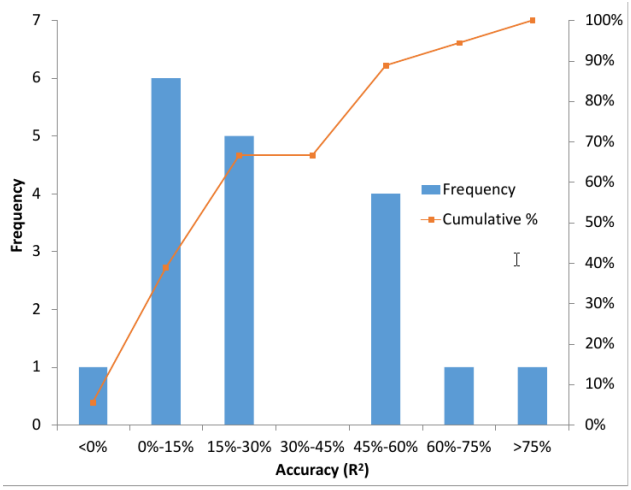


Fig. 6. The Macro-prediction approaches.

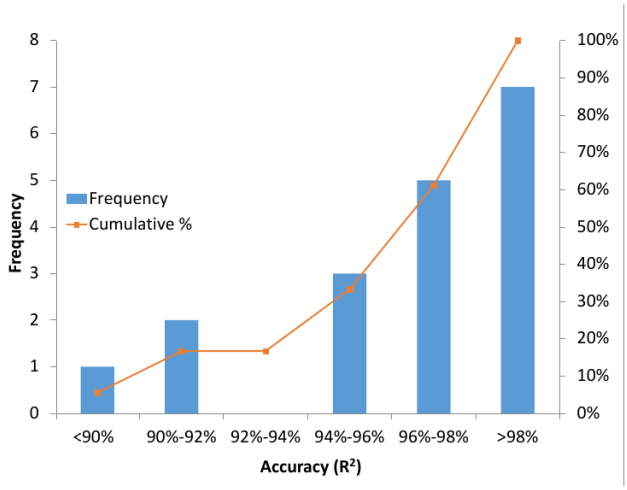
for forecasting SDSC Par's traces and around 124% for ANL-interpad. While our approach improved all the MAPE scores to below 6% (the majority to below 1%), notably in the traces of SDSC Par's and ANL-interpad (see Fig. 8).

TABLE VII. COMPARISON OF UNI-ATTRIBUTES FORECASTING

Forecasting approach	R^2	MAPE
LSTM-RNN	30%	42.78%
MICRAST	97%	2.38%



(a) LSTM-RNN



(b) MICRAST

Fig. 7. Comparison of R² results for uni-attribute forecasting between (LSTM-RNN and MICRAST).

The above results are mainly due to the characteristics of cloud traces, as demonstrated in the analysis illustration in Subsection II-B and Fig. 1. Some traces exhibited abrupt and unexpected variations with a high standard deviation. Specifically, the standard deviation of Requested Number of Processors in SDSC Par 1996, 1995, and ANL-Intrepid was above 10K. Without a suitable extraction process, the impact of such a characteristic poses a great challenge for the LSTM-RNN. This, in turn, led to notably low and unstable performance. While the performance of our approach suggests that the filtering and clustering processes were highly effective for extracting useful patterns from even these traces. For example, we observed this phenomenon with extracted patterns from the ANL-Intrepid trace (shown in Fig. 9) against their pre-

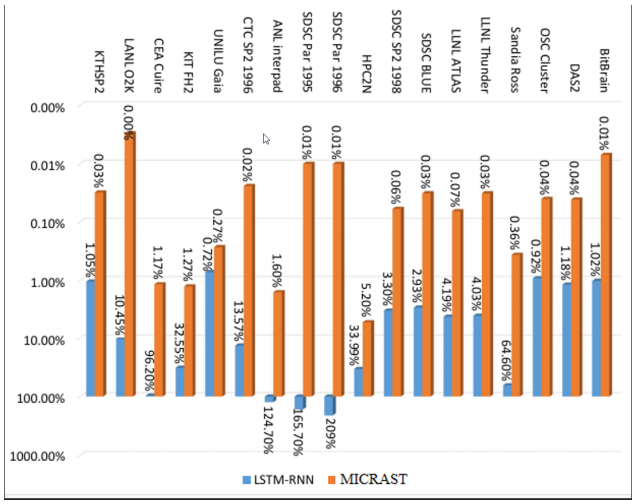


Fig. 8. Comparison of MAPE results for uni-attribute forecasting.

extracted versions (illustrated in Fig. 1). As mentioned earlier, in this context, these patterns represent the hidden trends in users' consumption records. This facilitated the learning and prediction process for the RNN model in MICRAST.

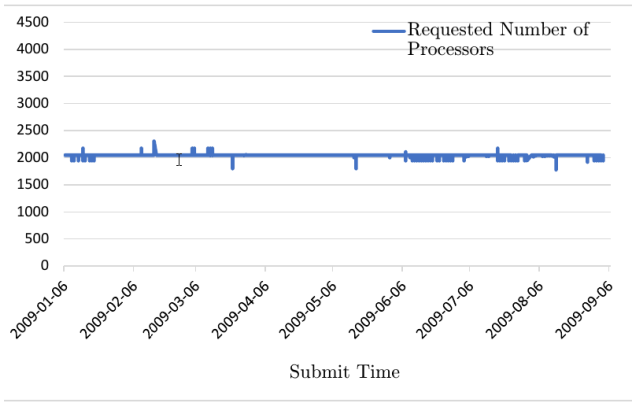


Fig. 9. Extracted pattern from ANLinterpad trace attribute.

Finally, we calculated the relative deviation (RD) for the R^2 results of both approaches. We drew boxplots for these RD distributions in Fig. 10. We compared them to show the level of consistency for each approach. Accordingly, the LSTM-RNN showed a wide range of RD spreading for 111 percentage points. While our approach performed with a narrower distribution for only 1.8 percentage points, showing more centered R^2 scores. Such narrow distribution with the high R^2 of 97% indicates that our approach can perform more accurate and consistent forecasting in the uni-attribute scenario compared to the related works approach.

2) Multi-attributes forecasting scenario: In the second scenario, we observed that related work exhibited even lower performance than previously. The cumulative distribution results in Fig. 11a show a low R^2 of below 5 percent for three of the traces. This is increased from only one trace in the uni-attribute forecasting. In contrast, our approach maintained its accuracy for the multi-attribute scenario, with no traces falling below 90% R^2, as depicted in Fig. 11b.

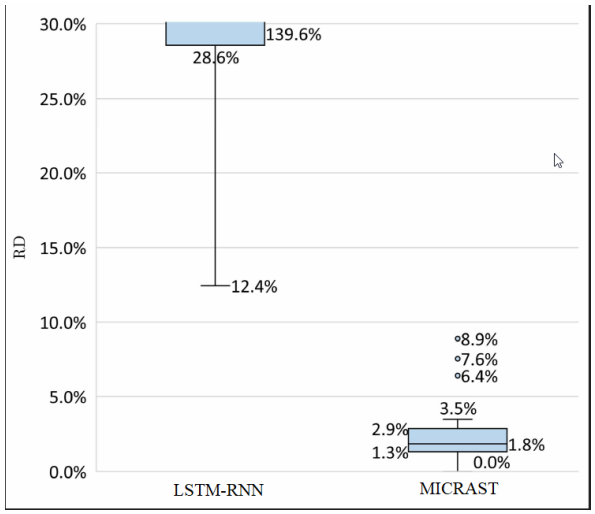
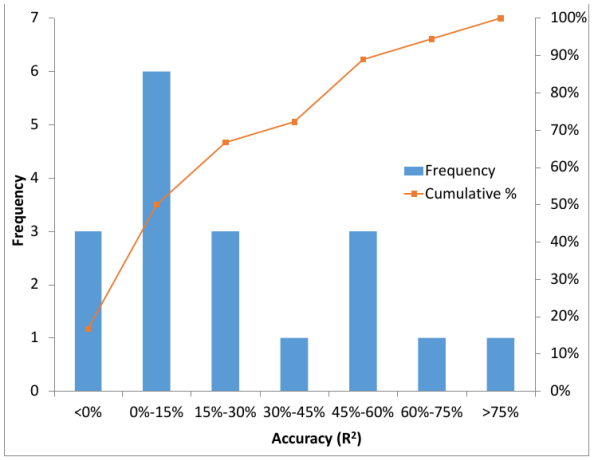
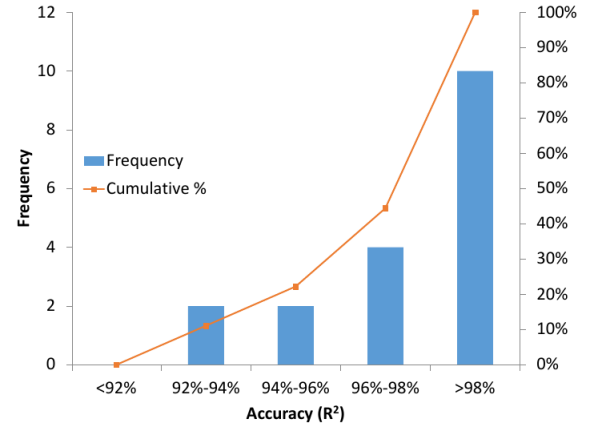


Fig. 10. A Relative deviation comparison for uni-attribute forecasting.



(a) LSTM-RNN



(b) MICRAST

Fig. 11. Comparison of R² results for multi-attribute forecasting between (LSTM-RNN and MICRAST).

Furthermore, the scores in Fig. 12 show an even higher MAPE for the LSTM-RNN approach. It recorded around 166% to 211% MAPE for SDSC-Par's traces and around 127% for ANL-Intrepid. This is an increase of about 2 percentage points compared to uni-attribute forecasting. While our approach maintained the MAPE of below 5.30% for all the traces (Table VIII).

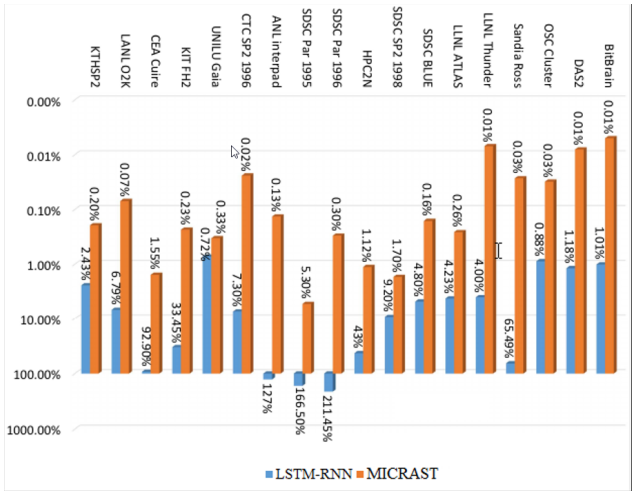


Fig. 12. Comparison of MAPE results for multi-attribute forecasting.

TABLE VIII. COMPARISON OF MULTI-ATTRIBUTES FORECASTING

Forecasting approach	R^2	MAPE
LSTM-RNN	27%	43%
MICRAST	97%	1%

These results are due to challenges caused by the use of multiple attributes with sudden changes characteristic. Such characteristics cause difficulties for the LSTM-RNN approach to capture possible correlation between these attributes, as they lack in providing meaningful patterns. While the extraction phase in MICRAST enables uncovering these attributes detailed patterns through clustering, making it easier for the prediction model (i.e. RNN model) to capture possible correlations.

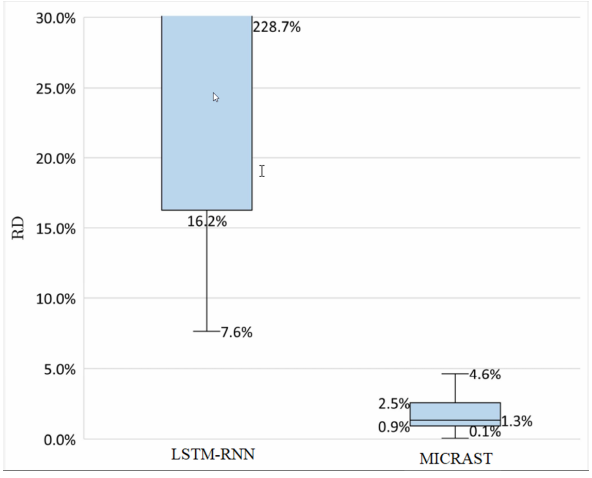


Fig. 13. A Relative deviation comparison for multi-attribute forecasting.

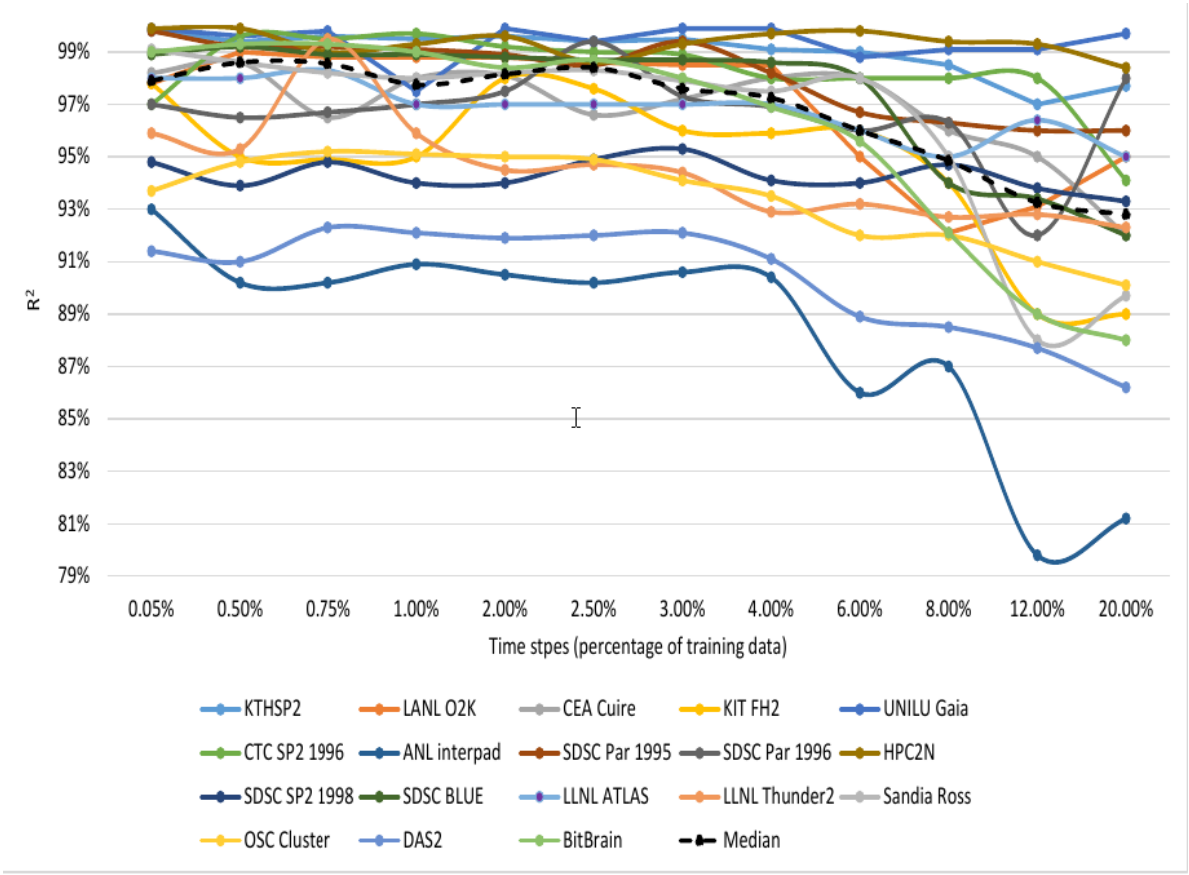


Fig. 14. Confidence range for MICRAST approach over time.

The boxplots for the relative deviation distribution of both approaches in Fig. 13 showed that LSTM-RNN failed to adapt to this type of forecasting. It recorded a relative deviation spread of 212 percentage points. This is higher than the uni-attribute forecasting by around 100 percentage points. While our approach maintains consistency in multi-attribute forecasting, the relative deviation spreads by only 1.6 percentage points.

B. Confidence Range for MICRAST

In this experiment, we measured the forecasting confidence by demonstrating the change in R^2 values for our approach as we extended the range of the forecast. We varied the range between 0.05% and 20% for each trace's training data (e.g. if the training data was 1 hour long, we made forecasts of 18s to 9m into the future). We have chosen this range because our observations showed that within this range there are significant chances for consumption pattern changes for each trace. Therefore, evaluating across the complete range demonstrates our ability to cope with forecasting even these changes.

We applied the same experimental configurations as in the previous evaluation. Similarly, we conducted uni-attribute forecasting of users' consumption preferences of Requested Number of Processors of all the selected traces in Table V. Finally, we calculated the median of these traces' R^2 for each step. Ultimately, the (R^2-median, R^2) over a particular forecasting range gives our MICRAST confidence.

The results in Fig. 14 show that our approach forecasted the majority of the traces with R^2 distributed at a range of 5 percentage points around the median of 98% R^2. This range expanded to 19 percentage points around the median of 93% R^2 when reaching 20% of the steps in the training data. This expansion is mainly noticed in the traces of DAS2 and ANL-Intrepid. As mentioned previously, these traces exhibit a significant characteristic of sudden changes in their consumption patterns, as illustrated for the ANL-Intrepid trace in Fig. 1 at page. This characteristic raises more challenges for the RNN model when the time step increases, even after the extraction process, affecting the prediction quality over time. Nevertheless, Fig. 14 shows that our approach can maintain the high R^2 median around 95% to 98% for the majority of the traces. While it drops by only 5 percentage points (to 93%) when reaching the full 20% of the rows from the training trace. This demonstrates that predictions up to 4% of the trace can be relied on for all traces, while for most traces we can reliably predict even 20% into the future of the training data.

V. Conclusion and Future Work

In this paper, we proposed an approach MICRAST for forecasting users' preferences in a cloud environment based on their consumption patterns. Our approach conduct this by extracting these patterns from the input traces through filtering

and clustering processes. Then it uniforms them through time alignment, linear interpolation, and normalisation. Finally, our approach passes the uniformed patterns for forecasting with RNN model, which we selected through a preliminary experiment. When comparing our work with prior art, we demonstrated that such extraction and pre-processing in MICRAST enables it to provide more efficient prediction for traces that exhibit characteristics of an abrupt change. We evaluated the MICRAST approach through the following experiments. First, we compared our approach against that used in the related works (i.e. LSTM-RNN) to demonstrate its superiority. Our approach showed the ability to conduct both univariate and multivariate forecasting with an accuracy of 98%, surpassing the LSTM-RNN approach by around 70 percentage points. Second, we measured the confidence range of our approach by observing how the accuracy changed when we increased how far ahead the forecasting needed to go. The results show that the MICRAST was able to forecast users' preferences with a confidence level between 95% and 98% when forecasting for a duration of 20% of the training data.

The limitation of our study is the lack of investigating it's benefit for real world application and it's efficiency for other types of application beyond cloud computing. Therefore, for future work, we aim to investigate the applicability of our approach for energy awareness improvements among private cloud users. After predicting users' consumption preferences, we could notify them for alterations they could do to their consumption. We also consider different scenarios of users' reactions and test the effect of these reactions on cloud utilisation by implementing them in cloud simulators such as CloudSim or DISSECT-CF. In addition, we intend to investigate our approach for other purposes and datasets besides cloud computing. One potential implementation of MICRAST is in energy management sectors, especially smart grids. By collecting the consumption records of different users in the grid, our approach could be used to extract and predict their patterns. This could help in managing demands, optimising grid operations, and planning renewable energy integration.

REFERENCES

[1] M. Seyedan and F. Mafakheri, "Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities," *Journal of Big Data*, vol. 7, no. 1, p. 53, 2020.

[2] M. Mehta, G. Pancholi, and A. Saxena, "Organizational resilience and sustainability: a bibliometric analysis," *Cogent Business & Management*, vol. 11, no. 1, p. 2294513, 2024.

[3] Y. S. Patel and J. Bedi, "Mag-d: A multivariate attention network based approach for cloud workload forecasting," *Future Generation Computer Systems*, vol. 142, pp. 376–392, 2023.

[4] Y. Lu, J. Panneerselvam, L. Liu, Y. Wu *et al.*, "Rvlbpnn: A workload forecasting model for smart cloud computing," *Scientific Programming*, vol. 2016, 2016.

[5] B. Feng, Z. Ding, and C. Jiang, "Fast: A forecasting model with adaptive sliding window and time locality integration for dynamic cloud workloads," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1184–1197, 2022.

[6] S. M. Ali and G. Kecskemeti, "Clustering datasets in cloud computing environment for user identification," in *2022 30th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. IEEE, 2022, pp. 165–171.

[7] ——, "Efection: Effectiveness detection technique for clustering cloud workload traces," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 202, 2024.

[8] ——, "Sequal: An unsupervised feature selection method for cloud workload traces," *The Journal of Supercomputing*, pp. 1–19, 2023.

[9] C. Chatfield, *Time-series forecasting*. CRC Press, 2000.

[10] I. Svetunkov and F. Petropoulos, "Old dog, new tricks: a modelling view of simple moving averages," *International Journal of Production Research*, vol. 56, no. 18, pp. 6034–6047, 2018.

[11] Investopedia, "Simple moving average (sma) definition," Dec 2021, accessed on 2024-01-13.

[12] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice (3rd ed)*, 2023, accessed on 2024-01-15.

[13] R. H. Shumway and D. S. Stoffer, *ARIMA models*. Springer, 2017.

[14] K. Holden, "Vector auto regression modeling and forecasting," *Journal of Forecasting*, vol. 14, no. 3, pp. 159–166, 1995.

[15] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall International Editions, 1994.

[16] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[17] Baeldung, "Prevent the vanishing gradient problem with lstm," 2024, accessed on 2024-01-15.

[18] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[19] R. Nau, "Statistical forecasting: Notes on regression and time series analysis," 2020, fuqua School of Business, Duke University.

[20] WallStreetMojo, "Unit root tests - definition, types, examples, and advantages," 2021, accessed on 2024-03-03.

[21] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.

[22] Q.-X. Zheng, Y.-L. Wang, P. Lu, S.-L. Liu, Y. Zhou, and J.-G. Zheng, "Automatic time-shift alignment method for chromatographic data analysis," *Scientific Reports*, vol. 7, p. 3907, 2017.

[23] Mathful, "Linear interpolation: Definition, formula, & example," 2024, accessed on 2024-05-18. [Online]. Available: https://mathful.com/hub/linear-interpolation

[24] A. I. Magazine, "A guide to different evaluation metrics for time series forecasting models," 2021, accessed on 2024-03-03.

[25] S. Glen, "Absolute error: Definition, formula, examples," 2020, accessed on 2024-01-15.

[26] S. Allwright, "How to interpret mape (simply explained)," 2022, accessed on 2024-01-15.

[27] V. Profillidis and G. Botzoris, *Chapter 5—Statistical methods for transport demand modeling*, 2019.

[28] Y. Sfakianakis, E. Kanellou, M. Marazakis, and A. Bilas, "Trace-based workload generation and execution," in *Euro-Par 2021: Parallel Processing: 27th International Conference on Parallel and Distributed Computing*. Lisbon, Portugal: Springer, 2021, pp. 37–54, proceedings 27, September 1–3, 2021.

[29] Y. Lu, L. Liu, J. Panneerselvam, X. Zhai, X. Sun, and N. Antonopoulos, "Latency-based analytic approach to forecast cloud workload trend for sustainable datacenters," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 3, pp. 308–318, 2019.

[30] A. I. Maiyza, N. O. Korany, K. Banawan, H. A. Hassan, and W. M. Sheta, "Vtgan: Hybrid generative adversarial networks for cloud workload prediction," *Journal of Cloud Computing*, vol. 12, no. 1, p. 97, 2023.

[31] S. Arbat, V. K. Jayakumar, J. Lee, W. Wang, and I. K. Kim, "Wasserstein adversarial transformer for cloud workload prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 12 433–12 439.

[32] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," in *Procedia Computer Science*, vol. 125, 2018, pp. 676–682.

[33] J. Kumar, A. K. Singh, and R. Buyya, "Self directed learning based workload forecasting model for cloud resource management," *Information Sciences*, vol. 543, pp. 345–366, 2021.

[34] J. Panneerselvam, L. Liu, and N. Antonopoulos, "Inot-repcon: Forecasting user behavioural trend in large-scale cloud environments," *Future Generation Computer Systems*, vol. 80, pp. 322–341, 2018.

[35] P. Nehra and N. Kesswani, "A workload prediction model for reducing service level agreement violations in cloud data centers," *Decision Analytics Journal*, vol. 11, p. 100463, 2024.

[36] T. Szandala, "Review and comparison of commonly used activation functions for deep neural networks," *Bio-inspired neurocomputing*, pp. 203–224, 2021.

[37] S. Ouhame, Y. Hadi, and A. Ullah, "An efficient forecasting approach for resource utilization in cloud data center using cnn-lstm model," *Neural Computing and Applications*, vol. 33, no. 16, pp. 10 043–10 055, 2021.

[38] M. Xu, C. Song, H. Wu, S. S. Gill, K. Ye, and C. Xu, "esdnn: deep neural network based multivariate workload prediction in cloud computing environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 3, pp. 1–24, 2022.

[39] M. Daraghmeh, A. Agarwal, R. Manzano, and M. Zaman, "Time series forecasting using facebook prophet for cloud resource management," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.