University of Miskolc

Faculty of Mechanical Engineering and Informatics

Tools for Extracting and Forecasting Detailed Consumption Patterns From Cloud Traces

PhD Dissertation

Author: **Shallaw Mohammed Ali**
MSc in Software Engineering

Jozsef Hatvany Doctoral School of
Information Science, Engineering and Technology

Head of Doctoral School: **Prof. Dr. Jeno SZIGETI**
Academic Supervisor: **Prof. Dr. Gabor Kecskemeti**
Co-Supervisor: **Dr. Karoly Nehez**

Miskolc
2025

Contents

1 Introduction

Data is one of the most important assets for studying different aspects of applied computing. It provides the foundation for understanding complex phenomena and uncovering valuable insights. In cloud computing, workload traces have been widely used as rich data sources for supporting more efficient resource management, as shown in [1].

To gain the most benefit from these data, clustering and forecasting methods have been widely exploited as key techniques for data analysis [2]. Clustering helps to extract important information from cloud traces, such as detailed patterns of resource consumption in users' records [3]. These patterns are necessary for implementing crucial resource management tasks (e.g., resource provisioning). It enables a more insightful understanding and, consequently, better forecasting of future consumption. Therefore, developing tools for such extraction and forecasting will ensure the identification of potential challenges and opportunities associated with resource utilisation and management.

These tools benefit various studies. For instance, Kecskemeti et al. [4] demonstrated that users' behaviours have a great effect on maintaining the free and unconstrained availability of cloud resources. Therefore, they proposed offering these users virtual tokens (so-called engaging options) to improve resource efficiency. This mechanism would require analysis of users' patterns at a detailed level to target the engaging options more efficiently towards the desired behaviour. The analysis process for these studies needs to be unsupervised, as many cloud records show ambiguity in their users' labelling.

1.1 Research Problems

Much of the literature has provided tools for trace analysis based on extraction via clustering and forecasting such as [3]. However, these works have overlooked supporting such tools with a detailed consideration of the human aspect. Analysing human patterns at this level is crucial for efficient resource and energy management [5]. In this context, clustering-based studies in cloud computing have neglected two of the main factors that affect clustering quality: the selection of dimensions (attributes) and the methods of clustering [6]. Both must be carefully considered to ensure that the clustering is not only accurate but also useful for subsequent analysis or decision-making processes. This is particularly pertinent in cloud computing, where workloads are often characterised by large-scale, dynamic, and complex datasets.

Regarding attribute selection, many data analysis studies, such as [7] and [8], have exploited methods of feature selection and dimensionality reduction. According to [9], the use of such general feature selection methods requires supervisory inputs (e.g., predefined labels and categories). These inputs are not typically available in cloud workload traces. Meanwhile, to address clustering method selection, studies often rely on generic and unautomated techniques, which are not reliable for repeated tasks. In addition, these tools require full trace analysis. This is not applicable for cloud traces, since many of these traces contain up to one million lines of users' records, making the full analysis process costly and less efficient for detailed-level pattern extraction.

On the other hand, for forecasting, related research such as [10], [11], has presented various prediction models. However, the forecasting approach in these studies was designed to deal with consumption patterns as trends by performing predictions at an overall level, which we will refer to as macro-prediction in this dissertation. Thus, such an approach lacks the ability to capture users' hidden patterns from cloud traces and to forecast them at a detailed, micro-prediction level.

1.2 Research Aims

This thesis aims to address the following gaps in existing analysis tools:

- Attribute and method selection that enable the extraction of patterns from cloud traces at a detailed level.
- A micro-prediction approach capable of capturing and accurately predicting these detailed patterns.

To address these gaps, our research is divided into the goals that are outlined as follows:

- The study of tools for attribute and clustering method selection that are more efficient for the characteristics of cloud traces. These tools should perform such selection for both single and multiple attributes, as well as clustering methods, without the need for predetermined parameters. They should be automated and unsupervised, and should not require the analysis of full traces. These tools should enable clustering methods to produce segments of extracted detailed patterns.
- The study of forecasting approaches that support capturing and predicting detailed patterns from cloud traces. This approach needs to

integrate the extraction tools described above and apply separate pre-processing to produce trainable segments for each of the extracted patterns. This needs to be followed by separate training for each of these segments to generate a trained network for each. This enables more accurate micro-forecasting.

Accordingly, the overall goal of this dissertation is *to develop automated, unsupervised tools for extracting and forecasting detailed cloud patterns without requiring full-trace analysis or predefined parameters.*

2 Literature Review

2.1 Review of clustering in cloud computing

While aiming to extract useful information from cloud workload traces, many studies have investigated the benefits of clustering methods. For example, Yousif et al. [3] proposed enhancing cloud resource utilisation by clustering tasks in Google workload traces. This was done by clustering resource usage attributes such as CPU and hard disk usage using two clustering methods: K-means and density-based clustering. In addition, Gao et al. [12] developed a hybrid approach incorporating K-means clustering to improve the accuracy of resource provisioning, also using Google cluster traces.

Another application of clustering methods was conducted by [13]. In that paper, the researchers aimed to increase the accuracy of predicting parallel application runtimes to improve the utilisation of parallel computing systems. This was carried out using datasets from a parallel workload for model training by targeting the application runtimes in these data without considering their specific users. Furthermore, Patel and Kushwaha [14] proposed exploiting two clustering methods, namely K-means and Gaussian Mixture Models, to address the heterogeneity in the BitBrains trace caused by the diverse nature of workloads that cloud systems encounter. In this trace, resource usage attributes were targeted for clustering. The results showed better clustering quality for the Gaussian Mixture Model, while less running time was required for the K-means method.

The majority of the above studies applied cloud trace clustering to improve resource provisioning and utilisation. This is reasonable, as the human aspect was not required for most of these studies. However, human-centred applications, such as steering users towards energy-aware usage, would require the analysis and extraction of humans's behavioural patterns. Thus, it is vital to target the human aspect of workload traces with clustering. This necessitates a comprehensive investigation to test the applicability of clustering for such an implementation, which was not sufficiently discussed in these works.

In addition, the cloud traces in these studies can consist of several attributes. These attributes may not all be beneficial for clustering. To deal with the multi-dimensionality in clustering, researchers have used feature (attribute) reduction and selection methods. The authors in [15] presented a novel three-way clustering approach for feature reduction. It achieves this by dividing the targeted data into three regional groups for clustering. The method involves random sampling and feature extraction to introduce randomness and diversity, which makes the algorithm more robust. Such methods reconstruct the original data, which might not be beneficial for cases of behaviour extraction.

As demonstrated in the above studies, feature selection and dimensionality reduction mechanisms have already been used in various analysis and forecasting applications for cloud computing. The selection process in these applications was either generic or resulted in new features that may not be beneficial for extraction. Furthermore, these studies mainly target the resource aspect of the traces, where the data characteristics are less complex and the necessary selection parameters are known. Hence, the challenge of dealing with complex human patterns in the absence of these parameters was not encountered. In other words, using such general methods of feature selection requires further input parameters (e.g., the number of expected clusters). These parameters are not usually available in cloud workload traces (i.e., they do not disclose the number of users whose utilisation patterns the traces represent).

It is worth noting that the above investigations were conducted using only simple clustering methods (such as K-means). Thus, a wider range of clustering methods was not thoroughly investigated. Consequently, they overlooked one of the main factors affecting clustering quality, which is method selection. Few techniques have been developed to detect in advance the best method among a given set for clustering a particular trace. Most suggested tools are either generic or non-automated. In an attempt to provide a semi-automated approach, Barak and Mokfi [7] presented an MCDM (Multi-Criteria Decision Making) group model to select and rank clustering methods. In this study, six clustering methods were evaluated using five indices. Nevertheless, these processes and results are descriptive, limiting their suitability for automation, and their outcomes can be difficult to explain and interpret. From the above, we deduce that current techniques are inadequate for identifying high-quality clustering methods for cloud datasets.

2.2 Review of Forecasting in Cloud Computing

In the area of cloud computing, researchers have developed various forecasting models for different purposes. Most of these models specifically aim to address the challenges of dynamic resource management and scaling.

In [16], Lu et al. proposed a model called RVLBPNN to forecast workload trends based on historical data, combined with the workloads' level of latency sensitivity. Later, [17] presented an improved version of RVLBPNN by incorporating the K-means clustering method. This new version predicts future workload trends based on the history of response time characteristics for these workloads.

Kumar et al. [18] 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. The model predicts workload values based on previous samples. The authors also presented a similar forecasting approach in [19], embedding a self-directed learning process to predict future demand from cloud servers.

On the other hand, Qiu et al. [20] introduced an advanced large language model (LLM) tailored for energy forecasting tasks (e.g., electricity load, solar, and wind power). In this study, the authors aimed to address the challenge of what is called *hallucination*. In this context, hallucination refers to cases where forecasting models produce outputs that seem plausible but are not grounded in actual user behaviour. This occurs when applying unstructured and ambiguous data, similar to those presented previously, which we disregarded in our dissertation to avoid such a phenomenon.

It can be concluded from the above that a similar forecasting approach is followed by the models in these studies. This approach targets and macro-predicts the overall values and trends of workloads. Such a methodology is designed to treat users' patterns in the traces as a whole. Unfortunately, the traces in their raw form do not reflect any meaningful patterns for prediction. Thus, the current models lack an efficient mechanism to uncover and capture the diversity and variability of users' consumption at a detailed level.

3 Research Methodology

3.1 Extraction tools

This subsection provides a detailed description of our tools, which aim to ensure more efficient extraction through clustering and achieve the goals in Section 1.2. These tools are the non-supervisory method for attribute selection (SeQual) and a technique for clustering attribute combinations and method pre-detection (EFection).

3.2 SeQual: Attribute Selection Method for Clustering Cloud Workload Traces

This method selects a clustering attribute for each trace without asking for supervisory inputs or full trace analysis. It conducts this by first clustering samples for each attribute. The clustering process is repeated for each attribute for a predefined number of clusters, which ranges from 2 to 50, or any reasonable range defined by the user. We have used this range, as in the supervised traces where user identification is available, we have observed that it is rarely possible for a trace to exhibit more than 50 uniquely identifiable patterns.

Then, SeQual measures the quality of these clustering results at each point in the above range using an internal validation metric. For this purpose, we selected the Silhouette Coefficient (SC) metric, since our technique required a simple computation process. As a result, this forms a sequence of SC scores from 2 to 50 for each attribute. By drawing a plot of this sequence, a scale of quality is obtained for each attribute.

Figures 1 show a sample of such scale of SC quality for the most relevant attributes from sample trace (the ANL-Interpad 2009). Based on this chart, each of the attributes (Requested Number of Process, Requested Time, Run Time, Submit Time and Wait Time) shows different scales with respect to the silhouette quality.

For instance, the attribute of Requested Number of Processors shows a higher average of silhouette in comparison to the attribute of Submit Time. The figure also demonstrates noticeable peaks and troughs in the entire range of the measurement. In this context, the peaks and troughs represent irregular and sudden changes in clustering quality. Such behaviour can also be seen in the SC scales of other trace attributes.

Based on these observations, we concluded that any attribute showing a high mean of SC with a sharp peak and trough will potentially have a high ability to extract detailed consumption patterns. As a result, the attributes with such behaviour will be ranked higher for extraction, while those with gradually increasing or stable trends will be ranked lower. This was used to devise the proposed SeQual method for attribute ranking and selection.

We will describe the SeQual method in algorithm 3.2. In the first step, the user selects a trace T for attribute selection (Step 1). The trace T consists of a set of attributes $C = \{c_1, c_2, \ldots, c_m\}$, and each attribute c_n is processed

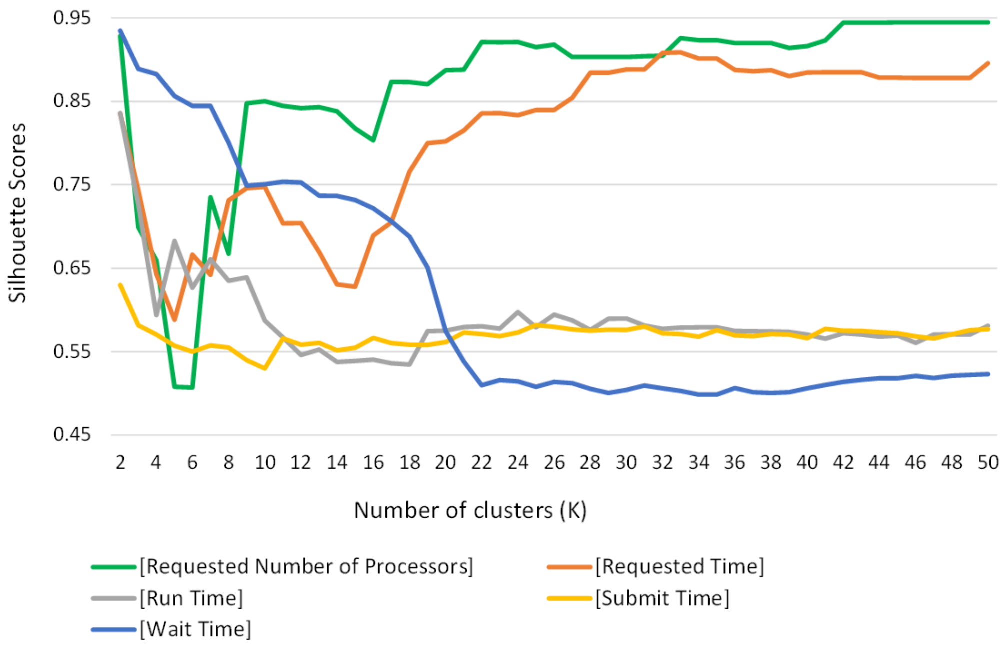


Figure 1: Behaviour of SC scores for all applicable attributes in the ANL-Interpad 2009 trace

individually. For each attribute c_n, the algorithm performs clustering using the K-means method, considering values for k ranging from 2 to 50 (Step 5). The results of the clustering process are stored in $\boldsymbol{C}_{n,k}$. After clustering, the quality of the clustering is evaluated using the silhouette coefficient $S_{n,k}$, which measures how well-separated the clusters are (Step 6).

This process of clustering and calculating the silhouette score is repeated for all values of k from 2 to 50 for each attribute. Next, we measure the $Quality(scale, average)$ for each attribute c_n by plotting the silhouette scores $S_{n,k}$ for each value of k and calculating the average of these scores for each attribute (Step 8).

Once the $Qulality$ $(scale, average)$ measured for all the attributes, they are analysed for ranking based on that. The attribute ranked highest if it exhibited $Quality(scale, average)$ with the sharpest peaks and troughs and the highest average of silhouette. A peak and trough are identified by finding a significant drop in the silhouette score over fewer than three clusters (Step 11). This entire process is repeated for each attribute c_n in the input trace T. Finally the ranked list of attributes along with their corresponding ranks are returned (Step 12).

Algorithm 1 The proposed SeQual method

Input: T, input traces; $C = \{c_1, c_2, \ldots, c_m\}$, attributes from T **Output:** Ranked attributes $[C, \mathbf{r}]$ with corresponding ranks.

```
Start
n ← 1
while n ≤ |C| do
    for k ← 2 to 50 do
        C_{n,k} ← Cluster(c_n, k)
        S_{n,k} ← Silhouette(C_{n,k})
    end for
    Quality (scale, average): Plot the silhouette score S_{n,k} for k = 2
  to 50 for each c_n and compute their average.
    n ← n + 1
end while
r ← Rank(C, Quality (scale, average))        ▷ Rank attributes based on
  silhouette score analysis (scale, peaks, troughs, and average)
Return [C, r]
Stop
```

As illustrated, the SeQual aims at addressing the challenges of ranking the best single attribute for clustering without the need for supervisory inputs or analysing the full trace. However, the dimensionality reduction aspect, where there is a need to detect which combination of attributes is best for the extraction, requires further consideration. Therefore, we propose the detection technique in the next subsection.

3.3 EFection: Effectiveness Detection Technique for Clustering Cloud Workload Traces

This subsection describes the process leading to EFection technique. This technique aims at offering the ability to detect the combination of attributes and the method that likely to give the highest clustering quality. It provide such detection without changing the original data, ensuring entire automation process.

3.3.1 EFection Description

The literature review above, showed importance of selecting an effective method and attributes to ensure high clustering quality. Algorithm 2 provides a detailed description of our technique, EFection. In this algorithm,

Algorithm 2 The proposed EFection technique

Input: (M, T)

Output: $k_{\max}, m$

1: $T_f \leftarrow$ Filtering T
2: $T_s \leftarrow$ uniform of random selection T_f
3: $D \leftarrow \emptyset$
4: $C_s \leftarrow \{\forall c_i, c_j \in T_s : R_2(c_i, c_j) > 0.5 \wedge c_i \neq c_j\}$
5: **for** $\forall m \in M$ **do**
6:     **if** $C_s \neq \emptyset$ **then**
7:         $K \neq P(C_s) \setminus \emptyset$
8:         **for** $\forall k \in K$ **do**
9:             $D \leftarrow D \cup \{(k, \mathrm{DB}(c_{\text{clustering}}(k, m)))\}$
10:         **end for**
11:     **else**
12:         **for** $\forall c_i \in D$ **do**
13:             $D \leftarrow D \cup \{(c_i, \mathrm{DB}(c_{\text{clustering}}(c_i, m)))\}$
14:         **end for**
15:     **end if**
16: **end for**
17: **Return:** $(k_{\max}, m : d \in D : \mathrm{DB}(c_{\text{clustering}}(k_{\max}, m)) = m_{\max})$

the user initially inputs the clustering methods and cloud trace into M and T, respectively. No further parameters are required, and our technique does not ask for a clustering validation criterion. By default, EFection considers all the attributes in that trace and all the selected clustering methods, but the user can limit the choice. Our technique's algorithm proceeds in three main phases:

- *Pre-processing phase*: The algorithm starts by conducting pre-processing in steps 1 to 3. It begins by filtering the provided trace T to T_f in step 1. The filtering process is conducted by disregarding attributes with a distribution of constant values above 80%, as they are deemed unsuitable for clustering. Then, our algorithm randomly selects the sample portion T_s from the filtered trace T in step 2. Each attribute in T_s is denoted by c. Each c is a vector of variables $(v_1, v_2, ..., v_n)$, essentially a column in the filtered trace, where n is the size of c. Sampling each attribute randomly ensures that the next phases will be less biased towards any particular part of the workload trace.

- *Analysis phase*: This phase extends from steps 4 to 16 to analyse the filtered attributes and the provided methods for potential detection. In this phase, the algorithm measures the R^2 for all combinations of pair attributes (c_i, c_j in T_s 4. If the R^2 between any pair of c exceeds 0.5, the corresponding pair is stored in C_s, which is a subset of T_s 4. As mentioned previously in the description of methodology, we used R^2 for filtering the combination attributes since it can determine a strong linear correlation between these attributes. Such a strong correlation can lead to an improvement in clustering quality when it reaches above 0.5, based on our observations. After identifying the combination of attributes that exhibit strong correlations, the algorithm performs an outer loop of the size of M 5, for each clustering method m in M. Within this loop, for each clustering method, if C_s is non-empty, the algorithm finds the power set of C_s (encompassing all potential attributes of c_i in C_s). These attributes are stored as subsets in the K set, excluding the empty set. This step gathers all possible combinations of attributes that are recommended. The absence of the empty set implies that there are no combinations selected, suggesting that clustering the attributes individually is preferable.

If there are recommended combinations (i.e., C_s is not empty), an inner loop 8 starts to measure the DB score for clustering each dimension of combinations in K with the method m 9. Each score, along with its corresponding dimension and clustering method, is catalogued in the set D. Otherwise, if C_s is empty, the inner loop of 12 measures the DB score for clustering each attribute c_i in T_s individually. These scores, along with their respective attributes and clustering methods, are then stored in the set D 13. In this phase, both R^2 and the DB metric work together to analyse the targeted samples. R^2 identifies which combination of attributes is nominated for clustering, while the DB metric assesses the internal validation for these attributes or their individual forms if C_s is empty. This analysis is performed separately for each of the selected clustering methods.

- *Detection phase*: Finally, upon completing the previous phase, the algorithm detects the optimal attribute and clustering method based on the DB scores. It performs the detection by returning the attribute subsets k_{max} and the clustering method m associated with the highest DB score 13. Essentially, the algorithm selects the clustering methods and attributes that yield the maximum DB score, which we have seen is likely producing better precision values.

3.4 MICRAST: Approach for More Efficient Forecasting

We propose the MICRAST approach to predict the future consumption patterns of cloud users. Our approach achieves this through a pipeline comprising segmentation, pre-processing, and forecasting, as illustrated in Figure 2. In this section, we cover both the training and forecasting phases of the approach, in comparison to existing methods.

Training Phase This phase proceeds as follows:

- The extraction pipeline employs clustering to uncover, from the input trace, the hidden patterns that drive users' requests. Based on our findings, clustering has demonstrated a strong capability for such extraction. To ensure efficient clustering, we perform two main tasks. First, we filter the trace by excluding attributes that hold the same value for more than 80% of the records. Such attributes are considered unsuitable for clustering, as illustrated in [21].

  Second, we apply two tools: the Sequential method of clustering Quality (SeQual) and the Effectiveness detection of clustering quality (EFection), to address both uni-attribute and multi-attribute feature selection scenarios. The SeQual method ranks individual attributes to determine the best candidate for extraction when the user opts for uni-attribute forecasting. Conversely, the EFection technique selects the most compatible combination of attributes for extraction in the multi-attribute forecasting scenario. Notably, if EFection selects only one attribute, it is recommended that the user opts for uni-attribute forecasting instead. Additionally, we use EFection to select the most suitable clustering method for the chosen attributes. The selected clustering method then groups similar historical usage records along with their submission times to form consumption patterns for each user. The output of this task is a set of segments representing detailed patterns.

- Parallel pre-processing pipelines prepare each segment of detailed patterns for prediction. In their clustered form, these segments exhibit non-uniform scales and formats that do not satisfy the requirements for effective data forecasting. Therefore, in these pipelines, we perform uniforming processes in parallel, separately for each segment, as depicted in Figure 2. First, the current time sequence for each segment is standardised into a consistent format across all traces. We also apply time alignment to synchronise these segments on the same time scales. Second, linear interpolation is implemented to address any missing records.

Subsequently, the data in each segment are normalised to a range between 0 and 1. This normalisation is essential for efficient forecasting, as cloud workload traces often exhibit data on vastly different scales—for example, the standard deviation of Requested Time compared to Used Memory varies significantly. Without normalisation, such disparities can hinder forecasting performance, whereas normalisation facilitates better compatibility with ANN forecasting models. The output of these pipelines is a set of uniform segments, each ready to be used as input for forecasting training.

- A parallel forecasting pipeline feeds the uniformed segments into the RNN model for training. It is important to emphasise that the RNN model, which demonstrated high performance in our comparative experiments. Once the RNN model is sufficiently trained, this pipeline produces a trained network for each segment, which is then stored and used later to forecast new input traces from the service provider's system.

  In addition, this pipeline also computes the average centroid for each segment. These centroids are stored alongside the corresponding trained networks to facilitate the prediction phase.

Prediction Phase In this phase, our approach follows the same segmentation and forecasting pipeline used in the training phase. As shown in Figure 2, the new input data are first clustered into segments of detailed patterns, after which the average centroid is calculated for each segment. These segments are then fed into the appropriate trained networks to predict future events. This is achieved by comparing the centroid of each new segment to the stored centroids from the training phase. When a match is found within a defined range, the corresponding trained network is selected and applied to the current segment for prediction.

In contrast to our approach, the methods adopted in recent cloud studies follow a singular prediction pipeline. These methods are designed to perform data preparation tasks, followed by forecasting, all on the full input dataset without considering any detailed segmentation. The prepared data are used to train a single forecasting model, which is then directly applied to the entire new input data during the prediction phase. This traditional pipeline is referred to as *Macro-prediction*.

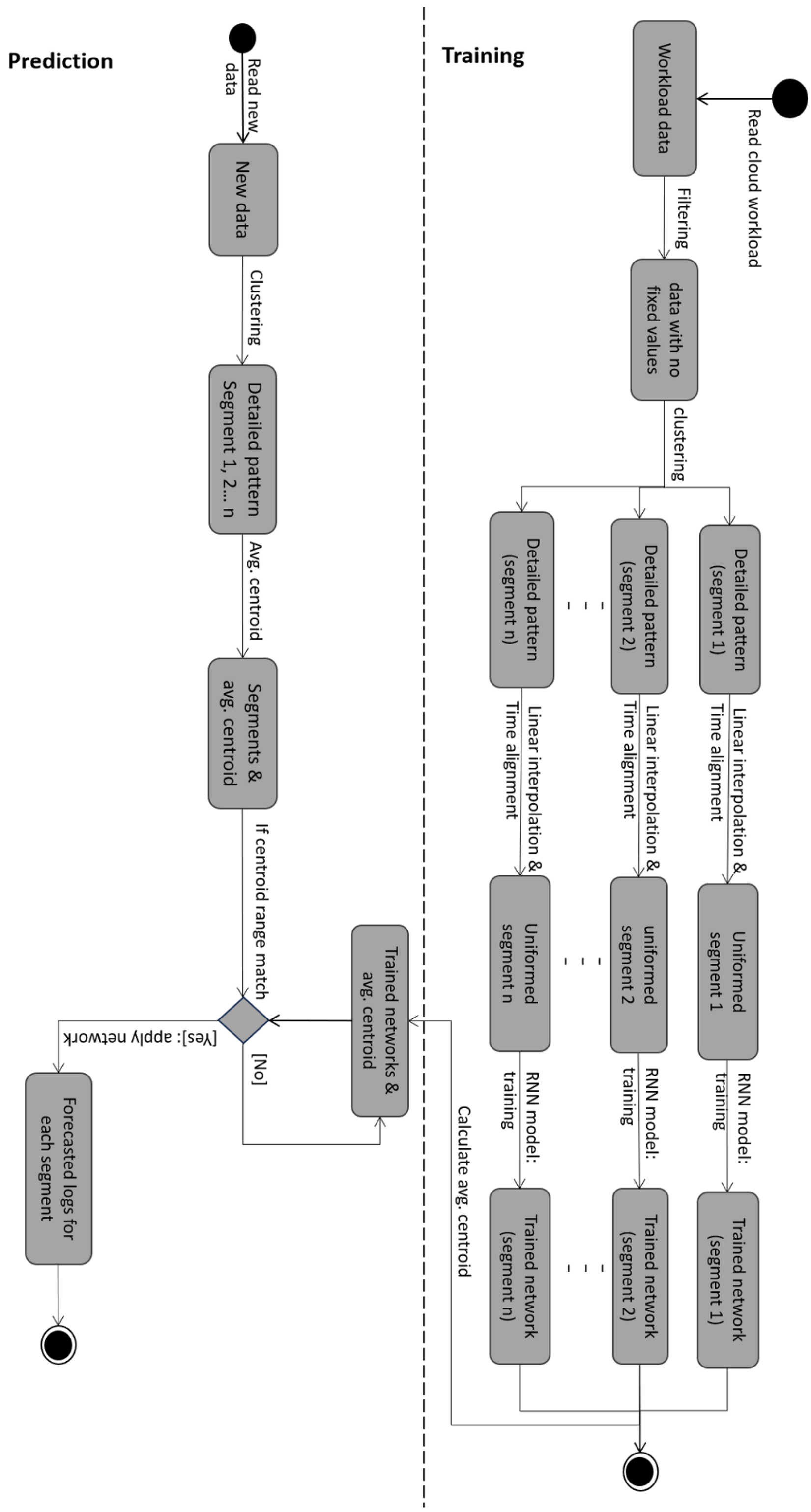


Figure 2: The MICRAST approach

4 Evaluation and Results

4.1 A Comparative Evaluation of SeQual

We begin with a comparative evaluation to assess the performance of the newly proposed SeQual method. This was carried out using 10,000 records randomly selected from the supervised trace attributes. The purpose of the SeQual method is to rank and select clustering attributes to extract detailed consumption patterns from cloud workload traces. Therefore, the evaluation criterion is based on each attribute's ability to produce meaningful patterns that can distinguish user labels via clustering.

To perform the comparison, we employed a ranking-difference approach. Each attribute was individually clustered in an attempt to reveal the corresponding users' labels. The clustering outputs were then evaluated against the user ID attribute using three quality metrics: Precision, Entropy, and Adjusted Rand Index. The closer the clustering results are to the actual user IDs, the higher the true rank of the attribute.

Next, we applied the SeQual method to rank the attributes and compared its output with three commonly used feature selection methods: Laplacian Score (LS), Random Forest (RF), and Principal Component Filtering (PCF). For this comparison, we used unsupervised traces.

For the supervised methods (PCF and RF), the user ID was used as the reference for generating the ranking. In contrast, for the unsupervised methods (SeQual and LS), the input attributes were evaluated without exposing any direct user identifiers (e.g., user IDs, user groups, or executable names). The effectiveness of each selection method was measured using the three metrics individually to avoid introducing bias from reliance on a single quality measure.

The overall comparison of SeQual with LS, PCF, and RF was performed using a weighting process. In this process, each attribute selected by a given method was assigned a weight based on the proximity of its clustering quality to the best-performing attribute, calculated as:

$$\text{Weighted selection} = \frac{\text{Clustering quality for selected attribute}}{\text{Highest clustering quality}}$$

In addition, we extended the evaluation by analysing the distributional characteristics of the supervised traces used for attribute ranking. Specifically, we examined two statistical measures relevant to clustering behaviour: the Coefficient of Variation (CV) and Skewness. We computed these measures for all attributes across the 19 supervised traces, then identified the characteristic ranges within which attribute rankings were most effective.

This analysis indicates a potential correlation between the quality of attribute ranking and these statistical properties. Accordingly, unsupervised traces that fall within similar ranges of CV and Skewness are expected to exhibit ranking performance comparable to those observed in the supervised traces.

4.1.1 Testing and Results

To present the performance of each method in greater detail, we displayed the ranges the input traces along with all performance metrics, in the boxplots of Figure 3. These boxplots illustrate the distribution of each feature selection method's performance. A method exhibiting the highest and least variable range of clustering quality across all validation metrics is considered to demonstrate more stable performance. Accordingly, SeQual showed the greatest stability among the methods tested.

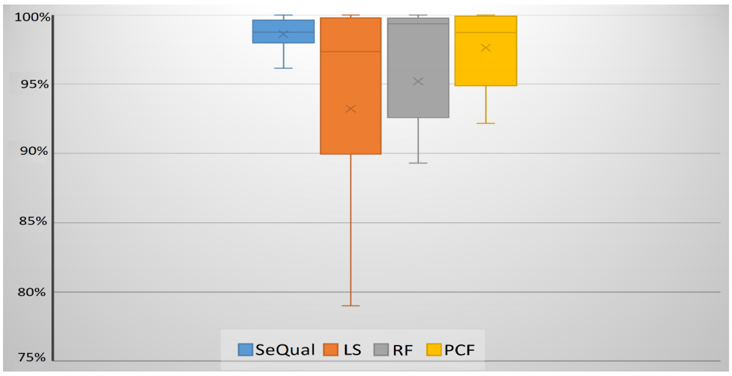
As shown in Figure 3, the SeQual method outperforms all other evaluated methods across the three quality metrics: Precision, Entropy, and Adjusted Rand Index. Specifically, according to the Adjusted Rand Index, SeQual achieves an attribute ranking accuracy of approximately 90%, compared to 74% for both LS and RF, and 79% for PCF. Regarding the Precision metric, SeQual attained a score of 99%, while the other methods achieved around 92%. Conversely, the Entropy metric exhibited less variation among methods, with SeQual scoring 99% and LS, RF, and PCF all achieving roughly 98%.

Based on these results, it can be concluded that despite being an unsupervised approach, SeQual demonstrates superior performance over the compared methods by margins ranging from approximately 8% to 28%. This improved performance is likely attributable to the fact that SeQual directly performs sample clustering during the ranking process, thereby capturing the underlying data structure more effectively.

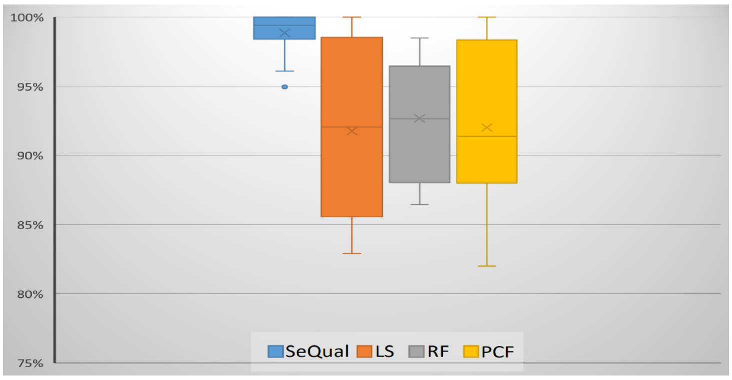
4.2 EFection Validation: Accuracy, Comparison, and Applicability

4.2.1 EFection Accuracy

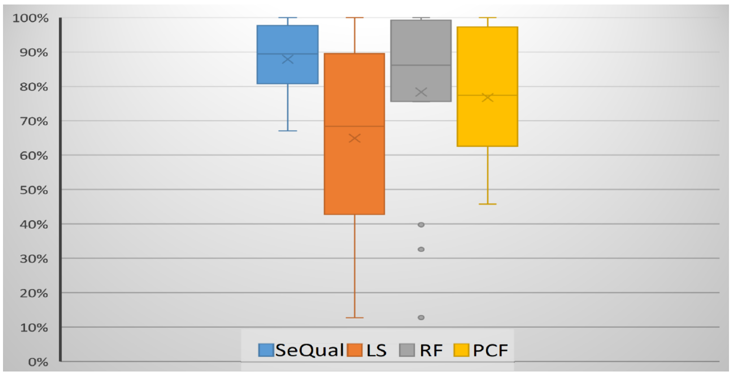
In the accuracy experiment, we used various combinations of clustering methods and attributes sampled from all the traces listed in Table 1. We began by measuring the precision of clustering for each of these samples. As in previous experiments described in this dissertation, clustering was performed



(a) Performance of methods based on Entropy



(b) Performance of methods based on Precision



(c) Performance of methods based on Adjusted Rand Index

Figure 3: Boxplots showing the distribution of methods' performance

with the goal of extracting users' labels based on their usage patterns, and precision was measured accordingly.

The attribute and method combination yielding the highest precision was considered the optimal choice. We then applied the EFection technique to

Table 1: Scenario for correct EFection suggestion

Clustering method	Attributes	MD clustering precision	Individual Clustering precision	EFection Suggestion
K-means	Wait Time	77%	65%	Run Time with F.First
	Run Time		81%	
F.First	Wait Time	91%	86%	
	Run Time		96%	

predict the best attribute-method combination for each input. To evaluate its performance, we calculated the overall percentage of predictions that matched the optimal choices. For the incorrect suggestions, we measured the deviation from the optimal selections and used these to determine the percentage error.

Based on these measurements, we calculated the overall accuracy of the EFection technique. We illustrated this experiment using two representative scenarios, as outlined below:

Scenario (1): Accuracy Measurement for Correct Detection In this scenario, we considered two attributes (Run Time and Wait Time) from the PIK-IPLEX trace and two clustering methods (K-means and F. First). We used the EFection technique to identify the best combination of attributes and method for clustering.

First, we measured the precision of the full trace for these attributes. As shown in Table 1, the precision for clustering Wait Time was 65% and 81% for Run Time when using the K-means method. By combining these two attributes, the precision dropped to 77%. While using the F. First method, the Wait Time recorded 86% precision and 96% for the Run Time. Similarly, the results of their combination dropped to 91%. By comparing these results for the F. First and K-means methods, we noticed that F. First showed higher precision than K-means, with a better result for individual clustering. These results showed that it was better to use Run Time individually with the F. First method for the clustering process.

Second, we compared the above results with the EFection suggestion. Our technique suggested clustering the attribute of Run Time individually rather than combining it with Wait Time when using K-means. Similarly, for F. First, it proposed clustering run time individually. For method comparison, EFection suggested using the F. First clustering method rather than

Table 2: Scenario for wrong EFection suggestion

Clustering method	Attributes	MD clustering precision	Individual Clustering precision	EFection Suggestion
K-means	Run Time	55%	58%	combine (Requested Time and Run Time) with K-means method
	Req. Time		57%	
F.First	Run Time	53%	53%	
	Req. Time		55%	

K-means. The comparison showed that the suggestion from our technique chose correctly the highest possible precision for the above scenario, in which the accuracy will be recorded as (96%/96%= 1) based on the following equation:

$$\text{Accuracy} = \frac{\text{suggested option precision}}{\text{highest possible precision}} \tag{1}$$

Scenario(2): Accuracy measurement for wrong detection This scenario considers two attributes of Run Time and Requested Time from the SDSC DS 2004 trace and both K-means and EM methods. The result in Table 2 showed that the precision was the highest, around 58%, for clustering Run Time individually with K-means. While EFection suggested using the attribute (Requested Time and Run Time) and the K-means method in the clustering process. In this scenario, we measured the difference between the precision of our technique's suggestion and the highest precision. Using the equation 4.2.1, the error percentage for this case recorded (55%/58%= 0.94).

Experimental results By applying the above evaluation methodology, the EFection technique was able to detect the best combination of attributes and clustering method with optimal choice (Accuracy = 1) in 83% of these cases. While the distribution of error percentages ranged between 2.8% and 10.8%.

4.2.2 Comparison with Recent Related Works

As previously mentioned, we introduced EFection as an automated technique for simultaneously detecting useful combinations of attributes and clustering methods. Most related works address these factors separately, offering individual techniques for each. To address dimensionality, Daraghmeh et al. [22] employed a PCA-based approach, while Barak and Mokfi [7] utilised an MCDM group methodology for method selection. Therefore, we evaluated EFection by comparing it with an integrated implementation of these two approaches (PCA & MCDM).

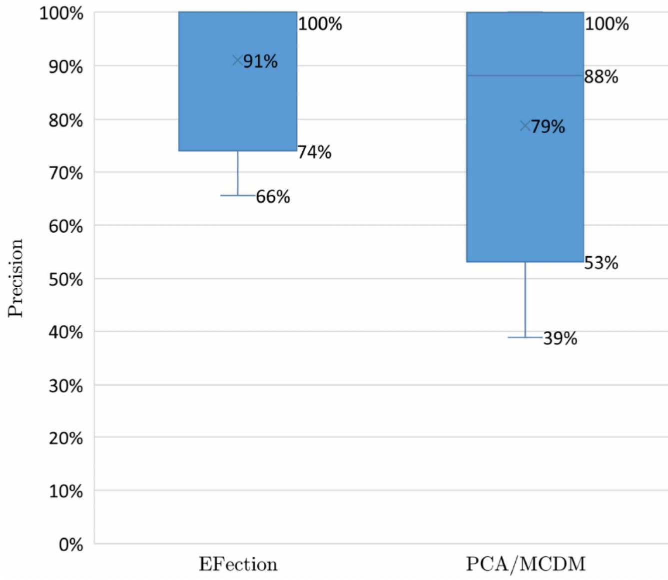


Figure 4: Comparison between EFection and PCA & MCDM performance

For the experiment, we used samples comprising clustering methods. Regarding PCA & MCDM, we first applied the MCDM methodology to select the clustering method, followed by PCA for dimensionality reduction of the attributes. Then, we applied EFection on the same samples. Similar to other experiments in this dissertation, we used the clustering process with the selected methods and attributes to extract users' labels and calculated the precision of the results. Finally, we compared the precision of the suggestions from EFection and PCA & MCDM with the combination that yielded the highest clustering precision, calculating how closely each suggestion matched the highest result.

The results, shown in Figure 4 as boxplots, reveal that EFection's precision ranged from 74% to 100%, with a median of 100%. Meanwhile, the PCA & MCDM approach achieved precision ranging from 53% to 100%, with a median of 88%. Additionally, EFection had an average precision of 91%, compared to around 79% for PCA & MCDM. This discrepancy is attributed to the PCA component generating new features, which reduced clustering

Table 3: Comparison between EFection's suggestions and attributes used in a utilisation improvement study

Clustering method	CT	NT	Reference
K-means (Euclidean)	85%	15%	Yousif and Al-Dualimy [3]
K-means (Manhattan)	82%	18%	Yousif and Al-Dualimy [3]
Density Based	66%	34%	Yousif and Al-Dualimy [3]
SOM	61%	39%	EFection suggestion

precision. These findings demonstrate that EFection offers better precision and stability compared to the combined PCA & MCDM approach, improving accuracy by 11 percentage points.

4.2.3 EFection Applicability in Clustering-based Studies

We evaluated EFection's applicability by comparing the clustering quality of its selections against the preferences in two related studies. We conducted this by employing our technique to select the best method for a utilization improvement study [3] and the most effective attributes for a pattern extraction study [14]. We selected these works as they align with the dataset characteristic of this thesis. Our technique's comparison against each study is illustrated as follows:

Compare Clustering Method Selection Against Utilisation Improvement Study The first experiment compared the clustering quality of the method suggested by our technique with the existing results in [3]. In their study, the authors employed K-means and density-based methods to cluster 12,500 records of Google workload traces, aiming at improving resource utilisation. For distance measures in K-means clustering, this work used Euclidean and Manhattan metrics. The number of clusters for these two methods was set at two. Two groups of attributes used in this test were computer tasks (CT) and non-computer tasks (NT). The selected attributes for CT included CPU rate, maximum CPU rate, cycles per instruction, and sampled CPU usage. For NT, the attributes were disc I/O time, local disc space usage, and maximum disc I/O time. To detect the best method, we adhered to the criterion that "The clustering algorithm, which divides the workload traces into two groups with an almost equal number of elements, is better to be applied" [3].

The results of using our technique showed that better results could be achieved by using the SOM method to cluster Google workload traces. By implementing this choice, SOM divided the traces into two parts, with a

Table 4: Comparison between EFection's suggestions and the methods used in a pattern extraction study

Attributes	No. clusters (Google trace)	No. clusters (Bitbrain)	Reference
CPU	13	15	Eva et al. [14]
CPU + Memory	18	15	Eva et al. [14]
CPU + Memory + Usage + Timestamp	21	17	EFection suggestions

proportion of 61% for CT and 39% for NT. This indicates that SOM was more effective than K-means and density-based methods in segregating the tasks of CT and NT, as illustrated in Table 4. This demonstrates that EFection successfully recommended a more efficient clustering method in this study.

Compare Attribute Selection Against Pattern Extraction Study In the second experiment, we compared the quality achieved by EFection's suggested attributes against the quality of the attributes used in [14]. In this study, Eva et al. investigated the extraction of characterisation and patterns of Google and Bitbrain workload traces based on CPU and memory utilisation. Two attributes were employed: CPU and Memory Usage. The paper utilised the elbow method to select the optimal predefined number of clusters. The study demonstrated that clustering results are more detailed when using the combination of CPU and Memory Usage compared to using them individually. The criteria employed were able to group the datasets with more detailed characterisation.

By applying the EFection technique, it suggests that even more efficient extraction (clustering) can be achieved by using the combination of CPU, Memory Usage, and Time-stamp instead of the original attributes, as presented in Table 4. The EFection suggestion resulted in 21 clusters for Google Trace and 17 for Bitbrain. This is more detailed compared to the original results, which were around 13 to 18 clusters for Google Trace and 15 for Bitbrain. This demonstrates the ability of EFection to detect better attributes than those used in related studies.

4.3 MICRAST vs LSTM-RNN for Related Work

In this evaluation, we compared the performance of MICRAST with the LSTM-RNN approach. This was conducted for both uni-attribute and multi-

Table 5: Comparison of uni-attribute forecasting results

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

attribute forecasting scenarios to ensure comprehensive validation. To measure each approach's performance, we used the R^2 and MAPE metrics. We selected these metrics because they provide a clear scale for measuring forecasting accuracy. They assess 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 proceeding to the results, we discuss the experimental configuration. Both forecasting scenarios use all the selected traces. Accordingly, we first utilised both approaches to predict consumption patterns for the selected attribute, which 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 trained the forecasting models with the historical records of an additional attribute (i.e., RunTime). Accordingly, we used the history of two attributes from the cloud trace to forecast the value of one particular attribute. We selected these attributes as they reflect major aspects of consumption (demand level and duration).

4.3.1 Uni-attribute Forecasting Scenario

Table 5 compares the average 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%, respectively. These results show a potentially significant improvement in accuracy when using our approach for uni-attribute forecasting.

4.3.2 Multi-attribute Forecasting Scenario

In the second scenario, we observed that the related work exhibited even lower performance than in the previous case. Table 6 shows the LSTM-RNN recored even lower R^2 dropped by 3 percent. While, our approach maintained it's high performance with 97% R^2 and only 1% MAPE.

These results are due to challenges caused by the use of multiple attributes with sudden change characteristics. Such characteristics make it difficult for the LSTM-RNN approach to capture possible correlations between these at-

Table 6: Comparison of multi-attribute forecasting results

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

tributes, as they fail to provide meaningful patterns. In contrast, the extraction phase in MICRAST enables the uncovering of detailed attribute patterns through clustering, making it easier for the prediction model (i.e., the RNN model) to identify potential correlations.

4.4 Confidence range for MICRAST

In this experiment, we measured 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 from 0.05% to 20% of each trace's training data (e.g., if the training data was 1 hour long, we made forecasts from 18 seconds to 9 minutes into the future). We have chosen this range because our observations showed that within this range there are significant chances of 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 patterns of Requested Number of Processors for all the selected traces. 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 Figure 5 show that our approach forecasted the majority of the traces with R^2 distributed within 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-Interpad. As mentioned previously, these traces exhibit a significant characteristic of sudden changes in their consumption patterns. 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, Figure 5 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

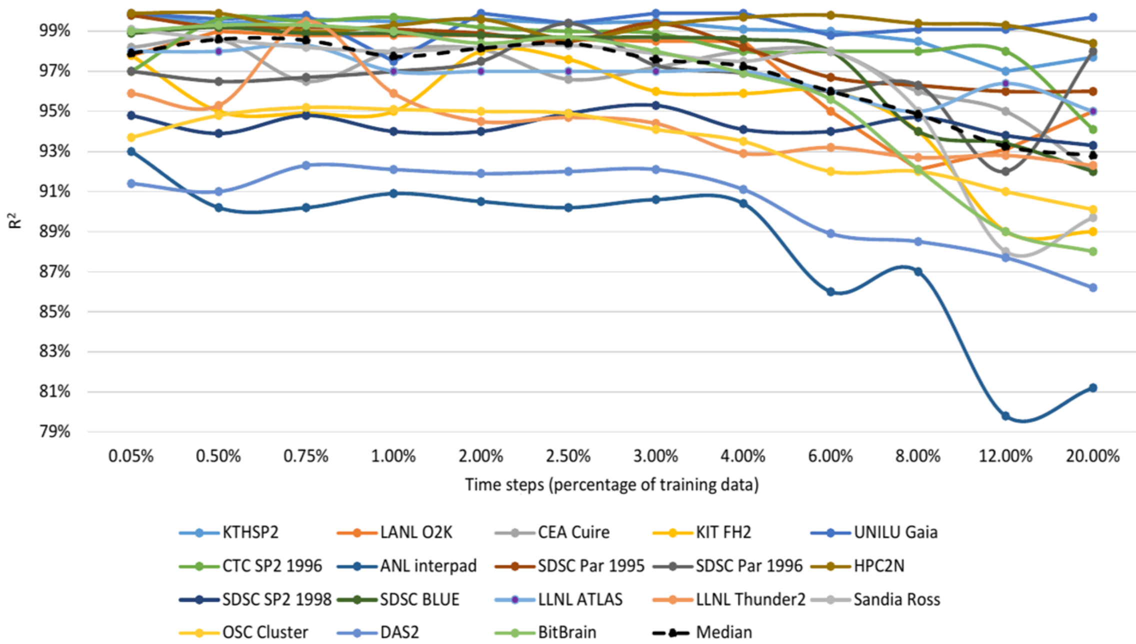


Figure 5: Confidence range of the MICRAST approach over time

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.

To wrap up, the evaluation and results in this section underscore the effectiveness of our extraction and forecasting tools. These findings highlight the robustness and applicability of our methods, setting a solid foundation for future research and development. In the next section, we will highlight these contributions and provide conclusions on their implications for future work.

5 Conclusion

5.1 Summary

In an attempt to obtain better cloud resource management, many studies have been used to extract and predict vital information from cloud records through clustering and forecasting tools. Such studies can assist greatly in steering users towards more aware usage if they target the patterns in these records at a detailed level.

Therefore, in Section 2, we initially conducted a through literature review. This review demonstrated the necessity of providing selection tools for clustering attributes and methods and developing an approach to forecasting that enables better detailed extraction and prediction of cloud patterns.

As a result, this thesis presented in Section 3 an analysis tools that provide the aforementioned abilities. For single-attribute selection, we proposed the SeQual method that ranks the candidate attributes for clustering by exploiting the ability of the silhouette coefficient metric. While, for multi-attributes and clustering method pre-detection, we developed EFection, which accomplishes this by using a combination of internal validation metrics (Davies-Bouldin) and the Coefficient of Determination. Whereas, for efficient prediction, we proposed the MICRAST which captures and predicts the detailed patterns from cloud traces. Our approach accomplishes this by wrapping up our previous extraction tool with phases of uniforming and time alignment.

The evaluation results in Section 4 demonstrated the performance of our extraction and forecasting tools. They showed that SeQual can compete with the supervised selection methods and perform better than unsupervised ones by around 8% to 28%. The results also supported the ability of the EFection technique to offer automated detection with high accuracy, around 83%, surpassing prior art by 15%. On the other hand, the assessment of MICRAST demonstrated its ability to forecast detailed patterns with a level of accuracy between 95% and 98%, outdoing related works by approximately 70%.

5.2 Contribution to Science

The new scientific findings of this dissertation are presented as follows:

- *My proposed method of clustering attribute selection, SeQual, performs more accurately without requiring supervisory inputs. It asks only for the data as input, making it more applicable to cloud traces that do not provide much information. To enable such selection, I performed sample clustering of the input attributes across a range of expected cluster numbers k. I then employed the silhouette coefficient metric to form a scale of quality scores for each value of k across these attributes. Finally, I examined this scale for the highest average silhouette score and identified the pattern of peaks and troughs to rank each attribute. The evaluation results support that my SeQual delivers higher accuracy compared to related methods by around 8% to 28%.*
  Related Publications: [P1], [P3], [P5]

- *My technique of pre-detecting multi-attribute combinations and clustering methods, EFection, operates without merging these attributes or* *involving manual intervention. By doing so, it preserves the originality of the information to be extracted and ensures its reliability for repeated tasks. To achieve such pre-detection, I utilised both the Davies–Bouldin index and the R^2 metric to analyse the quality of clustering samples for different attribute–method combinations. The validation results demonstrate that my EFection outperforms related works by approximately 15%.*

Related Publications: [P2], [P4]

- *My new approach to forecasting, MICRAST, micro-predicts cloud resource consumption by training forecasting models on pre-processed, detailed patterns extracted from cloud traces. It produces a trained network for each of the extracted patterns, which is then used to forecast the input trace. To support such prediction, I integrated both SeQual and EFection during the extraction phase, along with uniforming and time alignment for pre-processing. The validations of my approach MICRAST show that it achieves a confidence level between 95% and 98%, providing around 70% higher accuracy compared to existing macro-prediction approaches.*

  Related Publication: [P6]

5.3 Recommendations and Future Work

For future work, we have identified the following directions for our analysis tools. We plan to investigate whether the integration of other internal validation metrics, such as the Dunn Index, can enhance the accuracy of detection within our extraction tools, SeQual and EFection.

Furthermore, we plan to test our tools on more diverse types of datasets beyond scientific cloud traces (e.g., web applications, serverless cloud functions, IoT systems, or platform-specific services like Azure). This will support the applicability of these tools for more general use. It is expected that our tools will be applicable to these traces that exhibit general characteristics similar to those of the traces used in this thesis for validation. However, using such data may raise the possibility of prediction hallucination, as discussed in Section 2.2. To mitigate this, it is recommended that future systems incorporate a human-in-the-loop mechanism. In this approach, users or domain experts would have the ability to review forecast outputs, provide contextual input, or override model predictions when necessary. Introducing this form of human oversight could improve the reliability and contextual relevance of the predictions, while also supporting ethical alignment. This recommendation aims to reduce over-reliance on automated decisions and promote transparency, trust, and accountability in data-driven forecasting systems.

5.4 Publication Related to This Dissertation

[P1] Ali, Shallaw Mohammed, and Gabor Kecskemeti. ”SeQual: an unsupervised feature selection method for cloud workload traces.” The Journal of Supercomputing 79.13 (2023): 15079-15097. **Scoups indexed[Q2]**.

[P2] Ali, Shallaw Mohammed, and Gabor Kecskemeti. ”EFection: Effectiveness Detection Technique for Clustering Cloud Workload Traces.” International Journal of Computational Intelligence Systems 17.1 (2024): 202. **Scopus indexed[Q2].**

[P3] Ali, Shallaw Mohammed, and Gabor Kecskemeti. ”Clustering datasets in cloud computing environment for user identification.” 2022 30th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). IEEE, 2022. **Scopus indexed.**

[P4] Ali, Shallaw Mohammed, Gabor Kecskemeti. ”Cloud user Behavior prediction for resources usage awareness” In: Molnár, Dániel; Molnár, Dóra (eds.) XXIV. Tavaszi Szél Konferencia 2021: Absztrakt kötet Bp, Hungary : Association of Hungarian PHD and DLA Students (2021) 667 p.p. 401.

[P5] Ali, Shallaw Mohammed, Gábor Kecskeméti. ”The Prediction and Analysis of Cloud User Behavior for Resources Usage Awareness” In: Ivanyi, Peter (eds.) Abstract book for the 17th MIKLOS IVANYI INTERNATIONAL PHD and DLA SYMPOSIUM : ARCHITECTURAL, ENGINEERING AND INFORMATION SCIENCES Pécs, Hungary : Pollack Press (2021) 227 p. p. 106.

[P6] Ali, Shallaw Mohammed, Gábor Kecskeméti. ”MICRAST: Micro-forecasting approach for cloud user consumption pattern based on RNN” International Journal of Advanced Computer Science and Applications,, 2025. **Scopus indexed[Q3].**

References

[1] Z. Zhou, J. H. Abawajy, and F. Li. *Analysis of energy consumption model in cloud computing environments*. 2019.

[2] M. Kureljusic and E. Karger. Forecasting in financial accounting with artificial intelligence–a systematic literature review and future research agenda. *Journal of Applied Accounting Research*, 2023. ahead-of-print.

[3] S. A. Yousif and A. Al-Dulaimy. Clustering cloud workload traces to improve the performance of cloud data centers. In *Proceedings of the World Congress on Engineering*, volume 1, pages 7–10, 2017.

[4] S. Ostermann, G. Kecskemeti, and R. Prodan. Fostering energy-awareness in scientific cloud users. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 149–154. IEEE, 2014.

[5] Hong Huang, Yu Wang, Yue Cai, and Hong Wang. A novel approach for energy consumption management in cloud centers based on adaptive fuzzy neural systems. *Cluster Computing*, 27:14515–14538, 2024.

[6] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. d. F. Costa, and F. A. Rodrigues. Clustering algorithms: A comparative approach. *PloS one*, 14(1):e0210236, 2019.

[7] S. Barak and T. Mokfi. Evaluation and selection of clustering methods using a hybrid group mcdm. *Expert Systems with Applications*, 138:1027–1035, 2019.

[8] S. Chormunge and S. Jena. Correlation based feature selection with clustering for high dimensional data. *Journal of Electrical Systems and Information Technology*, 5(3):542–549, 2018.

[9] S. Solorio-Fernandez, J. A. Carrasco-Ochoa, and J. F. Martinez-Trinidad. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948, 2020.

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

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

[12] J. Gao, H. Wang, and H. Shen. Machine learning based workload prediction in cloud computing. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2020.

[13] J.W. Park and E. Kim. Runtime prediction of parallel applications with workload-aware clustering. *The Journal of Supercomputing*, 73(11):4635–4651, 2017.

[14] E. Patel and D. S. Kushwaha. Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia Computer Science*, 171:158–167, 2020.

[15] A. Li, Y. Meng, and P. Wang. Similarity-based three-way clustering by using dimensionality reduction. *Mathematics*, 12(13):1951, 2024.

[16] 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*, pages 37–54, Lisbon, Portugal, 2021. Springer. Proceedings 27, September 1–3, 2021.

[17] 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*, 5(3):308–318, 2019.

[18] 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*, volume 125, pages 676–682, 2018.

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

[20] Zihang Qiu, Chaojie Li, Zhongyang Wang, Renyou Xie, Borui Zhang, Huadong Mo, Guo Chen, and Zhaoyang Dong. Ef-llm: Energy forecasting llm with ai-assisted automation, enhanced sparse prediction, hallucination detection. *arXiv preprint arXiv:2411.00852*, 2024.

[21] Shallaw Mohammed Ali and Gabor Kecskemeti. Sequal: an unsupervised feature selection method for cloud workload traces. *The Journal of Supercomputing*, 79(13):15079–15097, 2023.

[22] M. Daraghmeh, A. Agarwal, and Y. Jararweh. An ensemble clustering approach for modeling hidden categorization perspectives for cloud workloads. *Cluster Computing*, pages 1–25, 2023.