Article

Not peer-reviewed version

Optimizing Multi-time Series Forecasting for Enhanced Cloud Resource Utilization Based on Machine Learning

Mateusz Smendowski * and Piotr Nawrocki

Posted Date: 9 September 2024

doi: 10.20944/preprints202409.0616.v1

Keywords: Cloud computing; Cloud resource usage optimization; Machine learning; Time series forecasting optimization; Cloud FinOps

 doi:10.20944/preprints202409.0616.v1

Article

Optimizing Multi-Time Series Forecasting for Enhanced Cloud Resource Utilization Based on Machine Learning

Mateusz Smendowski * and Piotr Nawrocki

AGH University of Krakow, Faculty of Computer Science, al. A. Mickiewicza 30, 30-059 Krakow, Poland
* Correspondence: alibaug143@gmail.com

Abstract: Due to its flexibility, cloud computing has become essential in modern operational schemes. How- ever, the effective management of cloud resources to ensure cost-effectiveness and maintain high performance presents significant challenges. The pay-as-you-go pricing model, while convenient, can lead to escalated expenses and hinder long-term planning. Consequently, FinOps advocates proactive management strategies, with resource usage prediction emerging as a crucial optimization category. In this research, we introduce the multi-time series forecasting system (MSFS), a novel approach for data-driven resource optimization alongside the hybrid ensemble anomaly detection algorithm (HEADA). Our method prioritizes the concept-centric approach, focusing on factors such as prediction uncertainty, interpretability and domain-specific measures. Furthermore, we introduce the similarity-based time-series grouping (STG) method as a core component of MSFS for optimizing multi-time series forecasting, ensuring its scalability with the rapid growth of the cloud environment. The experiments performed demonstrate that our group-specific forecasting model (GSFM) approach enabled MSFS to achieve a significant cost reduction of up to 44%.

Keywords: cloud computing; cloud resource usage optimization machine learning; time series forecasting optimization; cloud FinOps

1. Introduction

The increasing importance of cloud computing under- scores its undeniable flexibility, which renders it indispensable in modern organizations. However, the operational model inherent in cloud environments carries significant risks that can impact both cost-effectiveness and energy utilization. While the pay-as-you-go pricing scheme provides a convenient interface for accessing cloud resources, expenses can escalate uncontrollably [40].

Ensuring top-notch QoS (*Quality of Service*) and steering clear of SLA (*Service Level Agreement*) violations often involves provisioning more resources than necessary, leaving a considerable margin of unused cloud capacity. While overprovisioning leads to resource wastage and higher costs, under provisioning risks service downtime. Thus, FinOps (*financial operations*) emphasize finding a middle ground to encourage proactive resource management strategies [49].

Among the leading FinOps principles are continuous monitoring, embracing serverless architectures, utilizing preemptive computing instances, and employing resource reservations [49]. Since on-demand provisioning carries the risk of resource shortage, organizations can assess their workload patterns in a data-driven manner without compromising on performance and scalability. Moreover, long-term reservation plans can serve as the foundation for committed use discounts, further enhancing the substantial value generated by cloud environments [37].

As cloud computing continues to grow in scale and complexity, optimizing resource utilization is crucial for achieving environmental sustainability. Effective optimization can result in significant cost savings by reducing unnecessary computational expenses for both users and cloud service providers. Moreover, improving resource management aligns with the principles of GCC (*green cloud*

 doi:10.20944/preprints202409.0616.v1

computing) and sustainable computing by lowering energy consumption and reducing the carbon footprint of data centers, thereby facilitating economically viable cloud computing practices [44]. Given the rapid evolution of cloud environments, it is also essential that optimization solutions are scalable and do not introduce excessive operational overhead.

Therefore, resource usage prediction emerges as a significant optimization category. However, workload patterns in cloud environments are influenced by various factors, and as a result time series data represent historical resource usage characterized by complex, multi-seasonal dependencies and considerable variability [27].

Consequently, in this research, we introduce a novel system for data-driven cloud resource usage optimization called the multi-time series forecasting system (MSFS). While many studies tend to prioritize a model-centric approach [41], we propose a concept-centric attitude, emphasizing factors such as prediction uncertainty, interpretability and efficient data utilization in a knowledge-based manner. Recognizing the importance of scalability, especially given the fast-paced evolution of cloud environments, we focus on optimizing multi-time series forecasting through the introduction of similarity-based time-series grouping (STG) method and another related concept of group-specific forecasting model (GSFM).

Therefore, the major contributions of this paper can be summarized as follows:

- A novel cost-effective multi-time series forecasting system (MSFS) for dynamic cloud resource reservation planning.
- A context-aware multi-time series optimization method called similarity-based time-series grouping (STG) for scalable, automated, and feature-based time series segmentation.
- A novel hybrid ensemble anomaly detection algorithm (HEADA).
- A multifaceted evaluation of multi-time series forecasting (MSFS) system using a real-life dataset from a production cloud environment.
- FinOps-driven qualitative and quantitative assessment of dynamic resource reservation plans.
- The experiments conducted demonstrated that the group-specific forecasting model (GSFM) approach presented in this study outperformed both the global forecasting model (GFM) and local forecasting model (LFM) concepts, resulting in an average cost reduction of up to 44.71% in cloud environments.

The rest of this paper is structured as follows: Section 2 contains a description of related work, Section 3 focuses on multi-time series forecasting system (MSFS) architecture, Section 4 concentrates on the experiments performed, and Section 5 presents conclusions and further work.

2. Related Work

Cloud resource usage optimization covers a wide range of categories, but there has been a noticeable rise in the utilization of machine learning within this research domain. Consequently, the underlying premise of the analysis of related work is taking under consideration not only methods directly linked to cloud resource usage optimization (Section 2.2) but also recent advancements in time series forecasting (Section 2.1).

2.1. Time Series Forecasting

Nowadays, statistical-learning-based methods are less frequently the focal point of research; instead, they often serve as a baseline. In [29], the authors introduced the spARIMA (*self-paced auto regressive integrated moving average*) concept to address performance instability caused by noisy samples. Prior to applying the forecasting model, the authors employed a data difficulty ranking and then utilized the SPL (*structured prediction learning*) regime to gradually increase the complexity of samples used in training. Consequently, spARIMA outperformed both ARIMA and online ARIMA. Although statistical models offer significantly increased interpretability compared to machine learning and deep learning ones, they struggle to reveal nonlinear dependencies and multi-dimensional, complex patterns, which limits their applicability to data originating from cloud environments.

 doi:10.20944/preprints202409.0616.v1

In [13], the authors introduced a hybrid deep neural network for volatility forecasting. They expanded the data preprocessing stage by employing an encoding framework to transform one-dimensional time-series into GAF (*gramian angular field*) images, thereby enabling the subsequent application of a CNN (*convolutional neural network*). While deep learning may seem appealing since it bypasses the manual feature engineering stage, it requires large, reliable, and representative historical datasets. This necessarily limits the applicability of this approach in newly established, small or medium-sized cloud environments facing data scarcity.

RNNs (*recurrent neural networks*), alongside their variants encompassing LSTM (*long short-term memory*) and GRU (*gated recurrent unit*) cells, have become a state-ofthe-art solution for processing long sequences [56]. Despite the variety of architectures used for forecasting, researchers are increasingly focusing on the general stability of predictive modeling. Therefore, in [32], the authors introduced CNN-FCM (*CNN-fuzzy cognitive maps*) to address the scenario where samples in a dataset are not identically and independently distributed. Despite using latent feature extraction with neural networks and applying clustering algorithms (such as K-means and fuzzy C-means) on the embedded sequences, the major limitation of this approach is the lack of domain-specific metrics for automatically determining the optimal number of clusters. Only standard metrics were utilized, whereas domain-specific measures are critical in the cloud computing context.

In [59], the authors noted that conventional deep learning models often overlook the distinctive characteristics of time series data, particularly trends and seasonality. To address this, they developed an information fusion transfer mechanism designed to enhance the exchange of extracted long- term characteristics between data sequences. Despite improving forecasting accuracy by up to 60% compared to the Informer model by incorporating a sequence decomposition block as a pre-processing step in the novel DFNet (*decomposition fusion network*) architecture, which replaced the traditional self-attention mechanism, a major limitation of DFNet was performance degradation when handling irregular and noisy data with outliers. Subsequently, in [31], the authors concentrated on enhancing generalization abilities with machine learning and deep learning models under conditions of data scarcity. The researchers employed various data augmentation techniques, including adding random noise, sequence permutation, and scaling. This not only helped mitigate overfitting but also improved the performance of ResNet (*residual network*). However, a limitation of this study is the lack of evaluation and insights into how the hybrid approach – utilizing multiple data augmentation techniques simultaneously – performs on time series-specific architectures, such as LSTM or GRU networks.

In the domain of time series forecasting, jointly addressing interdependencies among multiple metrics brings superior outcomes. Consequently, in [6], the authors applied multivariate analysis alongside the implementation of the GFM concept [23]. Although this study aimed to explore the factors affecting overall GFM performance based on simulated datasets from various domains, the model selection criterion was based solely on evaluation metrics.

Appropriate domain-level measures that would allow for a comprehensive assessment of the solutions' properties were absent. Moreover, the conclusions drawn were related to the specific architectures implementing the GFM concept rather than to the GFM concept itself. The concept of GFM was further elaborated upon in [4], where the authors introduced an LSTM-MSNet (*LSTM-multi-seasonal net*) model designed for dealing with multiple seasonal patterns. However, evaluation metrics were found to be satisfactory only in the case of homogeneous datasets, which constitutes a limitation of this study in the context of cloud computing, where historical data from multiple virtual machines is heterogeneous.

In [5], the authors decided to follow the strategy of employing individual LSTM models for each set of similar time series. Common approaches to determining similarities among time series include distance-based, feature- based, and model-based methods [30]. Consequently, the researchers in [5] favored a feature-based approach, where extracted predictors were subsequently fed as input to algorithms such as K-means, DBScan (*density-based spatial clustering of applications with noise*), and PAM (*partition around medoids*). Unfortunately, the use of handcrafted features has two main drawbacks: it requires domain expertise and can result in predictors that negatively impact the

 doi:10.20944/preprints202409.0616.v1

model's performance. The authors did not conduct a study on feature importance or dynamic feature selection. Additionally, the manual feature enrichment is not scalable with the increase in the number of time series, as it necessitates expert intervention, and the ranking of features itself can change over time.

Leveraging low-dimensional data representations proves to be successful across various domains. In [35], the authors presented the unsupervised Signal2Vec method for universal time series representation. However, a limitation of this method is that it considers the entire time series rather than sequences, meaning that all data samples, including older ones, are given equal weights. This approach is not ideal for similarity analysis in cloud computing, where the most recent resource usage patterns are the most crucial. Furthermore, in [11], the authors introduced the HyVAE (*hybrid variational autoencoder*) approach, aiming to incorporate the learning of both local patterns and temporal dynamics of data sequences. Subsequently, in [27], the authors introduced the FEAT (*feature-aware multivariate time-series representation learning*) framework. FEAT utilized both timestamp-wise and feature-wise embeddings, combined with data augmentation techniques and a decoder layer to flexibly extract low-dimensional representations of signals. However, in [51], the authors found that a sequence- to-sequence LSTM network produced topologically correct embeddings of the time series sequences in the hidden space, effectively capturing the state of the underlying Rössler system, thus eliminating the need for a dedicated solution for semantic sequence representation. Unfortunately, all the aforementioned approaches conduct a comparative study of embedding models based solely on first-level metrics, such as reconstruction accuracy, while lacking domain-specific metrics that should support model selection and are introduced in our research.

As stated in [32], the features extracted by deep learning architectures generally outperform conventionally designed predictors. Accordingly, in [33], the authors aimed to optimize multi-horizon time series forecasting performance by leveraging stacked autoencoders to generate contextual embeddings. Those embeddings were then utilized in discriminative clustering. For each homogeneous cluster, the researchers applied a separate TCN (*temporal convolutional network*) model. Unfortunately, the researchers emphasized solely evaluation metrics, overlooking vital domain-specific measures in both the clustering and prediction phases. In our research, we aim to improve data efficiency and address the risks of direct forecast utilization. In [16], the authors addressed the quadratic time complexity of DTW (*dynamic time warping*), a representative distance-based similarity method. They introduced the first exact algorithm that made the running time dependent solely on the input coding lengths. Nevertheless, a limitation of the study is the application of the method in the context of time series similarity, where feature-based approaches are preferred for their semantic representativeness.

The third approach to time series similarity has been explored in [24]. The authors employed concept-based model extraction, matching time series to concepts, and conducting pairwise comparisons between concepts to determine similarities. However, this approach required creating additional models for each time series solely for the purpose of determining similarity, which introduces significant overhead.

In the domain of processing long sequences, the attention mechanism has emerged as revolutionary [48,52]. It not only forms the cornerstone of the transformer architecture, but also serves as an additional building block that can extend state-of-the-art models, such as LSTM networks [55]. However, complexity does not always lead to better results, both in terms of evaluation metrics and domain-level metrics. In [53], the authors highlighted a drawback of using the vanilla transformer model for multivariate forecasting – the inability to exploit spatial dependencies between variables. This limitation was addressed by the transformer-based LSTF (*long sequence time series forecasting*) model. Noticeably, researchers tend to focus on developing simpler alternatives, as demonstrated in [57], where multiple frequency-domain MLPs (*multi-layered perceptrons*) proved to be more effective learners than a single transformer. In the context of cloud environments, small models capable of quick adaptation appear to be particularly attractive. Frequent adaptations mean that cost-awareness becomes crucial for all processes related to the machine learning model lifecycle, such as retraining [34]. Cloud environments are indeed subject to rapid changes, and employing a model that lags in

 doi:10.20944/preprints202409.0616.v1

adapting to these dynamics could prove impractical. However, a common limitation of these approaches is their focus on high accuracy rather than on the trade-off between efficiency and responsiveness, neglecting the critical importance of

minimizing training time and model complexity for timely online adjustments.

2.2. Cloud Resource Usage Optimization

Optimizing resource utilization in a cloud environment is a highly intricate and multifaceted process. While our primary focus in this study lies in the application of timeseries forecasting, there exist equally sophisticated paths such as task scheduling optimization [10], virtual machine placement [47] and task allocation optimization [17]. In the realm of multi-objective approaches, various heuristics and genetic algorithms are commonly researched [10,22], along with their hybrid counterparts [19]. Despite the growing interest in machine learning-based optimization, it is essential to consider interpretability, which is a factor facilitated by employing rule-based systems [8]. In [9], the authors introduced an intelligent rule-based metaheuristic for task scheduling in time-critical applications. Cloud services are diverse, and almost any resource-intensive process can undergo refinement towards being more cost-efficient while maintaining the required high QoS and QoE (*quality of experience*).

Many cloud services incorporate autoscaling [45], where a reactive approach proves inadequate due to the time delay between underprovisioning detection and the provisioning of additional resources [15]. Consequently, a proactive approach leveraging time series forecasting with GRU network was examined in [58]. Furthermore, in [50], the authors enhanced the built-in Knative autoscaler by employing models such as ARIMA, LR (*linear regression*), LSTM, and BiLSTM (*bidirectional-LSTM*). This resulted in achieving both downtime minimization and a reduction in resource usage by 14%–20%. In [25], the authors introduced ProHPA (*Proactive horizontal pod autoscaler*), utilizing a BiLSTM network with an attention mechanism for multivariate workload forecasting. ProHPA demonstrated significant improvements, resulting in 23.39% and 42.52% reductions in CPU (*central processing unit*) and RAM (*random access memory*) utilization, respectively. This approach can perform poorly with new machines and those with diverse usage patterns due to the limited amount of data that supports scaling decisions. Consequently, a common limitation of the research on demand-based predictive autoscaling presented by the authors is its reliance on a dedicated model for each virtual machine – an approach based on the LFM concept – which goes against effective data usage. Additionally, the authors do not address mitigating the risks of incorrect forecasts, which is a focus of our research.

In [1], the authors introduced a sparse auto-encoder to retrieve low-dimensional workload representations. Subsequently, they employed a GRU network for CPU prediction over short-term horizons. This solution outperformed the RNN network in terms of learning stability. In [60], the authors introduced the entropy-optimized variational mode decomposition transformer – VMDSETformer. By decomposing the original time-series and assessing the complexity of each component separately using structured entropy, VMD- SETformer outperformed the LSTM network in the short- term prediction. Unfortunately, the time horizons evaluated may limit the ability to make informed operational decisions. Long-term forecasting proves to be much more useful in the context of dynamic resource reservation planning.

In [40], the authors introduced an SA-LTPS (*self-adapting long-term prediction system*) designed to optimize resource utilization for cloud-native applications. SA-LTPS comprised the RF (*random forest*) model for weekly predictions, enhanced by the PSO (*particle swarm optimization*) algorithm. The authors noted the inherent risk in forecasts; hence, an hourly co-routine was employed to monitor discrepancies between actual and forecasted usage. As a result, infrastructure costs were reduced by as much as 76–89% compared to scenarios without SA-LTPS and by 30–61% compared to those with an active autoscaling mechanism in Azure Cloud. Unfortunately, SA-LTPS required a running copy of the application to populate the sparse QoS table. Furthermore, SA-LTPS only covered univariate predictions, necessitating a dedicated system instance for each resource

 doi:10.20944/preprints202409.0616.v1

separately. While SA-LTPS addresses many critical aspects, its major limitations include scalability and high operational costs.

In [3], the authors presented a method for locally predicting scientific workflow runtimes; however, they did not provide or utilize it in the context of resource reservations. This omission prevents assessing the effectiveness of the solution in the context of FinOps. Furthermore, in [36], the authors leveraged long-term predictions to estimate resource demand in high-performance computing using XGBoost (*extreme gradient boosting*). This work was extended in [37], translating predicted demands into dynamic resource reservation plans, with neural networks and a TFT (*temporal fusion transformer*) included in the comparative study, which allowed for a 31.4% improvement in RMSE over the baseline model – Holt-Winters seasonal smoothing. However, both approaches required separate models for each scientific workflow or virtual machine, limiting their scalability in general-purpose cloud environments. Additionally, in [38], the authors highlighted the crucial role of exploratory data analysis in time series forecasting. They incorporated diverse methods for achieving multi-step prediction within the proposed model's architecture. Statistical tests and the analysis of multiple seasonal patterns provided valuable insights prior to the modeling process. However, a limitation of this study is the lack of context regarding FinOps and the estimation of resource reservation plans.

Data from cloud environments usually consists of a small fraction of outliers among a large number of time-series samples. Given the high cost of data labeling, the focus primarily shifts towards unsupervised anomaly detection methods. In [39], the authors concentrated on anomaly detection within the context of long-term resource usage planning. They proposed an algorithm called WHA (*weighted hybrid algorithm*) that combines the SMA (*simple moving average*), Kalman filter, and Savitzky-Golay filter for identifying

outliers. The evaluation was conducted using a prediction system composed of three modules: a metric collection module, an anomaly detection module, and a resource usage prediction module based on LSTM. The real-life test dataset included historical usage metrics from over 1,700 virtual machines. The results indicated that the WHA outperformed a static approach employing a LA (*limit-based algorithm*) and a DLA (*dynamic limit-based algorithm*) in terms of minimizing the average underestimation of anomalies. Furthermore, the method achieved a substantial cost reduction of up to 52.09% for Google Cloud services. In [39], the authors increased the confidence of classification through a weighted mechanism, but their study was limited to considering pointwise univariate anomalies. In contrast, our research considers hierarchical structures and also focuses on pattern-wise outliers, including multivariate ones. In [26], the authors achieved increased stability of verdicts through an ensemble-based algorithm called AERF (*adaptive ensemble random fuzzy*) to discover anomalous events during infrastructure operation and send them to the global event collector prior to such events. AERF was supported by the RFRB (*random fuzzy rule-based*) method. However, despite the dynamic weighted strategy it proposes, it is limited to labeled data only, which is a significant constraint on its implementation in the context of cloud environments. In [28], the authors introduced the FS-ADAPT (*few-shot time-series anomaly detection framework with unsupervised domain adaptation*) concept, which comprised two stages: a dueling triplet adversarial network and an incremental adaptation module. This framework addressed the target imbalance problem through few-shot learning, while unsupervised domain adaptation was employed to train models on data from one or more source domains and subsequently apply the acquired knowledge to unlabeled data from the target domain in operation. However, a limitation of this approach was the need for representative labeled data from multiple source domains. In contrast, our focus is on fully unsupervised methods for anomaly detection, aiming to enhance precision through hierarchical and weighted structures. In [2], the authors focused on unsupervised anomaly detection through a novel concept called USAD (*unsupervised anomaly detection for multivariate time series*). This approach featured a double autoencoder architecture within a two-phase adversarial training framework, which achieved a 24.09% increase in outlier detection accuracy compared to non- adversarial training regimes. However, a limitation of this approach is that, when attempts are made to integrate it into a lightweight data processing pipeline, maintaining the double autoencoder setup may introduce

 doi:10.20944/preprints202409.0616.v1

significant complexity. Furthermore, in [54], the authors introduced a novel unsupervised anomaly detection method targeting long-term seasonal patterns in data, called FCVAE (*frequency-enhanced conditional variational autoencoder*). However, a crucial limitation is that it is exclusively applicable to univariate time series. For more reliable detection of abnormal virtual machine conditions, multiple factors need to be considered simultaneously, which suggests that multivariate approaches might offer a more comprehensive solution. Despite this, the FCVAE outperformed baseline methods by up to 14.14% in terms of the best F1 score for univariate cases.

Counteracting inappropriate VM configurations and insecure allocations promotes the efficient use of cloud resources and infrastructure. Consequently, in [42], the authors proposed a novel MR-TPM (*multiple risks analysis-based virtual machine threat prediction model*) to proactively predict potential security threats related to virtual machine instances using the XGBoost model. Evaluated across various resource allocation policies, the solution achieved a reduction in cybersecurity threats by up to 88.9%. Additionally, MR-TPM incorporated workload prediction using a neural network-based approach. However, its evaluation was conducted using threat traces from sources such as Google Cluster, without focusing on scenarios of historical data scarcity, and thus excluding cases involving small or medium-sized environments. It also did not include a study on anomaly detection, which is an area covered by our research. Furthermore, unauthorized access to sensitive data contributes to excessive power consumption. Consequently, in [43], the authors proposed the ETP-WE (*emerging virtual machine threat prediction and dynamic workload estimation based resource allocation*) framework, which predicts both threats and resource usage in real time. Their approach achieved reductions in security threats, power consumption, and the number of active servers by up to 86.9%, 66.67%, and 30%-80%, respectively, while improving resource utilization by 60%-75%. However, the article lacked crucial FinOps context, as it did not provide estimated cost savings from the model's application or the costs associated with running the solution in real time in a cloud environment. In contrast, we advocate for incorporating a FinOps-driven approach. Similarly, in [20], the authors addressed the challenge of identifying malicious entities responsible for data misuse. They introduced a model based on a quantum neural network that predicts potential malicious data disclosures, effectively enhancing system security by up to 33.28% compared to similar solutions. In [21], the authors enhanced privacy preservation in cloud environments and improved model privacy accuracy by up to 15.89%, particularly for sensitive medical data. Additionally, in [46], the authors presented a novel PPMD (*privacy-preserving model based on differential privacy*) approach, which divides data into sensitive and non-sensitive partitions and injects noise into the sensitive segments. Further classification tasks employed in PPMD operation were evaluated using various machine learning algorithms including neural networks. Finally, the authors achieved an improvement of up to 16% in accuracy compared to similar security-oriented solutions.

2.3. Summary

Recent advancements in cloud resource usage optimization emphasize the significance of data-driven and machine learning-based techniques. Despite the considerable focus on leveraging diverse machine learning architectures for processing data sequences, there is a growing awareness of the

risk of inaccurate predictions, which necessitates improving their stability to ensure reliable decision-making.

Cloud FinOps principles include proactively shaping resource and cost management strategies. Consequently, concept-centric and model-agnostic solutions targeting enhanced resource utilization have become particularly crucial. Unfortunately, many studies, especially those focusing on deep learning, tend to assume access to large amounts of historical data, which may not be the case for early-stage cloud environments. Consequently, we observe a research gap in data-driven solutions that are applicable to both small and large environments, remain scalable during phases of dynamic growth and do not generate significant operational costs or management overhead.

Among the commonly researched approaches is the GFM, which might be too broad to fully capture the unique characteristics of all time series within the dataset. Conversely, optimization methods requiring a separate model for each virtual machine (LFM) could escalate costs as the

 doi:10.20944/preprints202409.0616.v1

environment expands. Additionally, while solutions aimed at identifying similarities between time series and applying predictive modeling to sets of similar sequences are being developed, these often encounter scalability limitations, overlook the key FinOps context, lack important domainlevel metrics and fail to pay increased attention to the contextual application of multivariate forecasts for longterm decision-making.

Furthermore, there is a notable trend towards favoring smaller models that can swiftly adapt to the variable conditions. This preference is understandable, especially considering that the models achieving the best forecasts in terms of evaluation metrics may not always lead to the best resource reservation plans [37]. However, it is worth noting that predictive autoscaling, which typically leans towards the use of smaller models, may overlook the possibility of cloud resources being unavailable during provisioning requests. Consequently, embracing forecasting within the framework of dynamic resource reservation not only facilitates more informed decision-making and enhanced budget planning but also supports demand-based and FinOps-aware resource management, while simultaneously minimizing the risk of resource shortages.

Additionally, in Table 1 we highlight some of the most important features in the context of this research and categorize the articles that exhibit these features. Among these features are machine-learning based, statistical learning-based, clustering enabled, anomaly detection aware, resource reservation, FinOps aware, and forecasting optimization (with a focus on increased stability and scalability of the process rather than solely achieving better accuracy).

Table 1. Feature analysis.

Feature	Articles
Machine learning-based	[4–6,11,13,23,27,31,32,51,56,59], [1–3,10,17,24,25,28,32,34,36–40,42,47,48,50,52–54,57,58,60], [20,21,43,46]
Statistical learning-based	[29,36–38]
Clustering enabled	[5,23,29,32,33]
Anomaly detection aware	[2,26,28,36–40,54]
Resource reservation	[37,39,40]
FinOps aware	[37,40]
Forecasting optimization	[5,24,32–34,53,57]

3. Multi-Time Series Forecasting System

In this section, we introduce a novel concept called the multi-time series forecasting system (MSFS) for FinOps- aware resource optimization (Section 3.1). Additionally, we present a formal notation to further explain key design decisions and showcase the challenges of time series forecasting in cloud computing that are targeted by MSFS (Section 3.2).

3.1. System Overview

The primary objective of enhancing cloud resource usage is to strike a balance between reducing costs, minimizing resource wastage and improving long-term operational decisions. Consequently, MSFS utilizes time series forecasting to enable demand-based resource scaling and reservations. Given that data from cloud environments encompass multiple multivariate time series of varying lengths and dimensions, our focus is on optimizing the time series forecasting itself to ensure overall MSFS scalability.

In general, MSFS involves adopting a hybrid approach between the global forecasting model (GFM) and local forecasting model (LFM) concepts. While the GFM proposes a shared forecasting model for all virtual machines, the LFM approach necessitates a separate entity for each virtual

 doi:10.20944/preprints202409.0616.v1

machine. Therefore, the hybrid solution – group-specific forecasting model (GSFM) – seeks to find a compromise between these extremes, offering scalability and improved accuracy in dynamic resource reservation. GSFM is thus a concept involving separate model forecasting for each group of similar time series, which is identified through similarity- based time-series grouping (STG) – a multi-time series optimization method. Consequently, MSFS is a self-adapting system that enables data-driven decisions, influencing the state of the cloud environment by creating dynamic resource reservation plans and delegating scaling recommendations to the resource manager.

Therefore, Figure 1 illustrates the concept of integrating MSFS within a cloud environment. The system relies on diverse metrics related to virtual machine resource usage, collected from cloud monitoring. This monitoring system is a core built-in component of cloud environments, which is responsible for aggregating data on the operational status of various infrastructure resources. Given the potential for numerous virtual machines within a cloud infrastructure (as illustrated in Figure 1, where there are M virtual machines), each with its own set of metrics (Figure 1 also shows the scenario where each of the M machines has K registered metrics), the system processes a dataset consisting of multiple time series. Within this framework, the MSFS system functions as a data consumer, utilizing historical data to forecast future resource usage. Simultaneously, MSFS acts as a data producer, generating resource reservation plans – essentially resource allocation recommendations for each virtual machine – based on the forecasts made. Subsequently, reservation plans serve as the basis for scaling decisions, which are delegated to a resource manager, thereby directly influencing the cloud infrastructure, for instance recommending adjustments to the computational capacity of individual virtual machines.

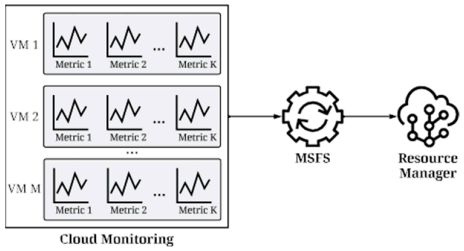


Figure 1. The concept of integrating the MSFS within a cloud environment.

Considering that as a solution that enhances resource utilization, MSFS should not incur higher costs or lead to increased management overhead, system architecture (Figure 2) has been designed as a set of eight stages, each with its individual execution schedule. Consequently, each stage can be executed as a serverless pipeline, and the execution of individual stages can be supervised by any workflow scheduler. In Figure 2, each stage contains a single step (highlighted block) indicating the starting point for on- schedule execution. If off-schedule execution occurs, the interactions depicted with dashed lines are followed.

The initial stage of MSFS – *Data Processing* – interfaces with cloud monitoring to retrieve resource usage metrics. This stage encompasses two modules: the *Ingestion Module* and the *Anomaly Module*. The former is responsible for applying lightweight data transformations, primarily to address erroneous readings and missing observations, while also performing streamlined data aggregations to unify dimensionality across time series. Moreover, the *Anomaly Module* is used for the detection and handling of outlying samples and anomalous sequences, leveraging the novel hybrid ensemble anomaly detection algorithm (HEADA), which is discussed in Section 4.1. Given that the *Anomaly Module* accesses historical data already devoid of erroneous readings, its primary objective is to detect time points where virtual machines encountered anomalous operation conditions, thereby ensuring the stability of classifications. Consequently, anomaly detection is integrated into the MSFS system through a dedicated *Anomaly Module*. In our proposed operational model, as outlined in the MSFS architecture, the HEADA algorithm plays a central role in the anomaly detection process, serving as an integral component of the system.

Throughout various stages, the steps of data loading and extraction into either TSDB (*time series database*) or VDB (*vector database*) occur. The use of MSFS in production environments is particularly

 doi:10.20944/preprints202409.0616.v1

recommended in order to ensure persistence, idempotency and facilitate backward filling. Additionally, it is worth noting that updating the computing capabilities of virtual machines often requires turning them off. To counteract this, downtime prevention strategies are possible, such as provisioning a new machine in advance. In such cases, it is necessary to associate MSFS-specific identifiers with cloud-specific ones to unambiguously distinguish virtual machines. Moreover, we introduce a concept of ID-SET as a collection of unique TSIDs (*time series identifiers*) for which the execution of specific stages takes place. This abstract collection is exchanged between stages during off-schedule execution and by default includes all the series identifiers in on-schedule runs. For instance, in the event that new time series are identified during the final step of the *Anomaly Module* (series meeting the minimum length criterion – determining the necessary data threshold for series inclusion in MSFS), these can be seamlessly added to the ID-SET collection, prompting the immediate initiation of the subsequent stage.

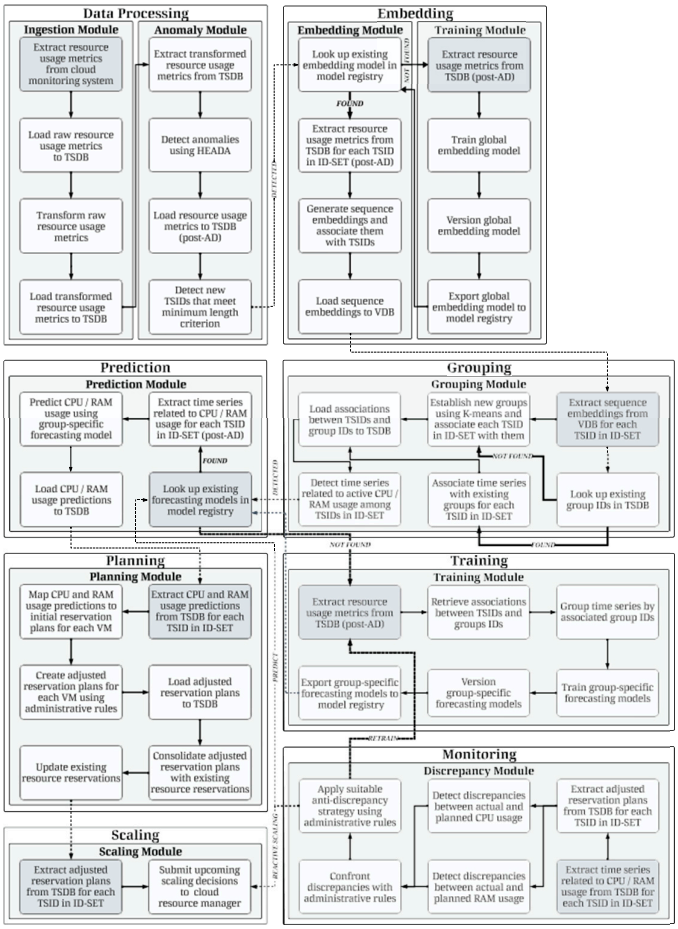


Figure 2. Diagram of the MSFS operation.

The second stage of MSFS – *Embedding* – is responsible for training the embedding model (e.g., autoencoder) within the *Training Module*, which is incrementally fine-tuned in subsequent stage executions. Such model is expected to efficiently capture nonlinearity and temporal dynamics in order to generate semantically accurate latent representations of sequences intended for feature-based similarity determination. Consequently, the design decision to implement a global model involves incorporating data from all available time series (post-AD – after anomaly detection). This design decision prioritizes overall effectiveness in sequence reconstruction over exceptional performance in corner cases. Furthermore, the *Embedding Module* is responsible for creating vectorized representations of sequences belonging to individual time series considered at this stage.

 doi:10.20944/preprints202409.0616.v1

Leveraging the embedding model to generate semantically accurate predictors reduces the need for handcrafting features. As the global embedding model undergoes periodic fine-tuning, the maintenance of feature importance ranking becomes obsolete.

In the subsequent stage of MSFS – *Grouping* – embedded representations of sequences are subjected to K- means clustering. With semantically accurate embeddings, this algorithm clusters together time series exhibiting similar resource usage patterns – this forms the foundation of the STG method. Additionally, if the dataset contains metrics that deviate significantly from the norm (for instance representing a virtual machine that has been accidentally left without decommissioning and thus with near-zero resource usage), STG can isolate such series, effectively acting as a filter. This prevents such series from adversely affecting other series, thereby enhancing further modeling outcomes and facilitating the identification of unused resources. Section 4.2 covers the selection methodology for the embedding model and focuses on K-means clustering, including the evaluation and determination of the appropriate number of clusters with the use of novel domain-specific metrics.

According to execution schedules, the subsequent stage of MSFS is *Training*, which directly leverages the associations of time series with groups. Following the introduced optimization strategy – STG – at this stage the *Training Module* aims to prepare group-specific models, thus implementing the GSFM concept. Subsequently, within the *Prediction* stage – and specifically as part of the *Prediction Module* – forecasting is performed exclusively for CPU and RAM (pertaining solely to active virtual machines), as these two resources determine the configuration parameters of virtual machines in a cloud environment. Consequently, predictions are made using group-specific models. Section 4.3 focuses on the evaluation of weekly forecasts for multiple virtual machines using GSFM, compared with GFM and LFM- based approaches, alongside a comprehensive assessment against domain-specific metrics.

In the subsequent stage of MSFS – *Planning* – forecasts of resource consumption in percentage terms are translated into initial resource reservation plans, specifying recommended vCPUs (*virtual CPUs*) and RAM over time for each active virtual machine included in MSFS. Recognizing the inherent uncertainty in forecasts and the potential impact of inaccurate predictions – leading to either SLA breaches or significant resource wastage – we introduce the concept of administrative rules. These rules form a set of policies guiding the refinement of initial reservation plans into adjusted ones, proactively minimizing the repercussions of incorrect data-driven recommendations. An example of administrative rules might entail adjusting initial vCPUs or RAM values if the recommendations surpass 80% of the machine's resources, as this could result in performance degradation. Consequently, administrative rules also serve as a means to inject domain knowledge of cloud platform administrators, when time series data is limited. Since the resource usage characteristics of virtual machines can change dynamically, historical data from a single machine may not be sufficiently representative for effective forecasting. Accurate predictions require a robust historical database to leverage past information and detect similar future events with high accuracy. Moreover, LFM overlooks potential interactions between virtual machines that may belong to the same cluster or communicate with one another, resulting in an erroneous assumption that they are isolated and independent. Thus, LFM can be seen as a significant example of inefficient data utilization. On the other hand, GSFM improves upon the GFM approach by grouping similar or related time series instead of using all available data indiscriminately. By grouping time series with similar characteristics, GSFM avoids the oversimplified assumption underlying GFM that resource usage patterns across all machines have a uniform constructive influence. GSFM employs knowledge-based decisions when constructing datasets for training group- specific models, addressing data insufficiency issues that affect LFM. In contrast, GFM aims for broad applicability but may struggle with highly diverse and chaotic usage patterns that a single model may fail to capture effectively. Thus, GSFM's approach of distributing knowledge across several representative models (serving as experts in specific resource usage characteristics) is more effective than relying on a single model that strives to be adequate for all possible scenarios and patterns.

 doi:10.20944/preprints202409.0616.v1

Table 8. Back-to-back evaluation metrics for resource usage prediction at the virtual machine level in the test set.

Co	ntext		Virtual Machine							
Evaluation Metric	**Resource**	**Method**	**VM01**	**VM02**	**VM03**	**VM04**	**VM05**	**VM06**	**VM07**	**VM08**
MAE	CPU	GFM	0.123	0.209	3.282	1.653	3.863	2.656	1.510	2.787
		GSFM	0.025	0.249	2.788	1.674	3.552	2.882	1.479	2.775
		LFM	1.019	0.308	2.966	2.379	12.511	4.156	4.503	2.910
	RAM	GFM	1.057	0.288	0.850	1.173	1.152	1.012	0.841	0.633
		GSFM	0.216	0.072	0.724	0.617	1.284	0.994	0.475	0.704
		LFM	8.764	0.285	1.024	4.270	2.291	1.270	3.712	1.364
RMSE	CPU	GFM	0.123	0.370	3.700	1.993	4.335	3.066	1.758	3.405
		GSFM	0.031	0.369	3.205	2.046	4.109	3.208	1.773	3.414
		LFM	1.022	0.396	3.390	2.698	13.027	4.743	4.785	3.429
	RAM	GFM	1.061	0.296	0.938	1.213	1.368	1.160	0.884	0.789
		GSFM	0.264	0.078	0.846	0.658	1.481	1.178	0.523	0.836
		LFM	8.793	0.288	0.179	4.336	2.426	1.333	3.761	1.489
MdAE	CPU	GFM	0.128	0.085	3.155	1.460	3.819	2.438	1.453	2.290
		GSFM	0.022	0.149	2.652	1.391	3.180	2.910	1.314	2.441
		LFM	1.045	0.260	2.857	2.420	13.348	4.792	4.752	2.424
	RAM	GFM	1.102	0.298	0.823	1.217	1.052	0.911	0.869	0.529
		GSFM	0.189	0.071	0.670	0.617	1.210	0.859	0.473	0.635
		LFM	8.984	0.274	0.980	4.363	2.368	1.288	3.784	1.419
FR	CPU	GFM	1.000	0.383	0.269	0.363	0.352	0.441	0.351	0.436
		GSFM	1.000	0.792	0.397	0.443	0.369	0.315	0.458	0.465
		LFM	1.000	0.818	0.402	0.182	0.038	0.150	0.022	0.523
	RAM	GFM	1.000	0.000	0.327	0.000	0.405	0.351	0.000	0.349
		GSFM	1.000	0.000	0.426	0.000	0.324	0.373	0.000	0.281
		LFM	1.000	0.000	0.843	0.000	0.133	0.286	0.000	0.077

However, the noted superiority of GSFM does not change the fact that predictions are subject to high uncertainty, and in the face of dynamic changes – resource usage characteristics previously unobserved in historical data – they may become significantly inaccurate. Therefore, proactive and reactive mechanisms are critical to minimize the negative effects of forecast inaccuracies. In MSFS, these mechanisms encompass administrative rules, adjusted reservation plans and various anti-discrepancy strategies. Given that the *Prediction* stage is just one component of MSFS, all pre-prediction and post-prediction mechanisms play a crucial role. Therefore, optimizing multi-time series forecasting should extend beyond solely improving prediction evaluation metrics to mitigating the risks associated with GFM and LFM, ensuring cost-effective model maintenance and addressing the critical FinOps factor – the scalability of undertaken actions and strategies. These objectives remain consistent with the properties of the GSFM approach introduced in MSFS.

4.4. Dynamic Resource Reservation Planning

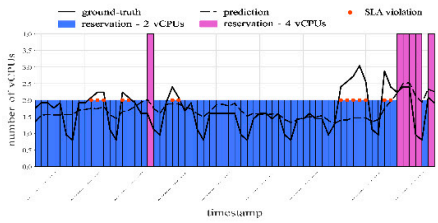
Dynamic resource reservation plans are derived from mapping resource usage predictions obtained from group- specific forecasting models – GSFM, providing recommendations for the

 doi:10.20944/preprints202409.0616.v1

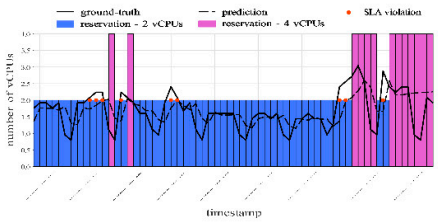
specific configuration of virtual machines. Therefore, we introduce the concept of a reference virtual machine type – a machine of this type would be statically reserved throughout the entire evaluation period, regardless of resource utilization.

This approach enables the creation, and subsequent assessment, of diverse resource reservation plans. Consequently, the reference machine types selected for the evaluation of the *Planning* stage of MSFS are general-purpose machines available on the Google Cloud Platform within the e2standard flavor. Therefore, when the reference type is set as c2d-standard-16, it is assumed that the forecasted usage percentage corresponds to the number of vCPUs and gigabytes of RAM associated with that specific machine type – 16 and 64, respectively.

Therefore, Figure 15 illustrates the consolidated initial CPU reservation plans for VM07 (which experienced the most SLA violations), with c2d-standard-16 set as the reference for forecast translation. Without MSFS enabled, this type would be used throughout the entire test set period. However, with MSFS, both dynamic reservations and scaling based on predicted demand are possible, resulting in significant resource savings. Additionally, a critical factor influencing the quality of initial reservation plans is the appropriate choice of the execution schedule for the *Prediction* stage. In the case of weekly predictions (Figure 15a), forecasts are considerably less accurate and lead to more SLA violations. With dynamic time series, more frequent decisions such as selecting the daily execution schedule enable greater accuracy and an improved balance between resource utilization and service availability (Figure 15b). With multiple forecasts per timestamp, the newer one overrides the older, resulting in possible updates of reservation plans. The mapping methodology presented for translating predictions into resource recommendations remains the same for RAM usage, as determining the recommended reservation type requires knowledge of both CPU and RAM demand.



(a) Weekly execution schedule of the *Prediction* stage



(b) Daily execution schedule of the *Prediction* stage

Figure 15. Initial CPU reservation plan for VM07 with e2-standard-16 as the reference virtual machine type.

As a result, the assessment of MSFS's *Planning* stage involved preparing initial reservation plans for each of the eight virtual machines in the dataset, followed by evaluating their characteristics across various selected reference types. Among the metrics applied are both quantitative costeffectiveness and qualitative indicators. Table 9 provides a summary of initial reservation plans based on weekly CPU and RAM predictions made with a daily execution schedule throughout the test set. Qualitative and quantitative metrics are averaged across all virtual machines to draw general conclusions.

Table 9. Averaged qualitative and quantitative indicators for initial reservation plans based on various reference virtual machine types.

Reference type	Percentage cost reduction (with MSFS)	Daily USD cost (without MSFS)	Daily USD cost (with MSFS)	Percentage CPU usage (with MSFS)	Percentage RAM usage (with MSFS)	Scaling events	Violation events	Percentage availability
e2-standard-2	0.00	2.07	2.07	17.69	22.13	0.00	0.00	100.00
e2-standard-4	39.40	4.14	2.51	35.39	38.17	0.63	0.13	99.81

 doi:10.20944/preprints202409.0616.v1

e2-standard-8	48.24	8.29	4.29	48.88	44.40	1.13	0.63	99.04
e2-standard-16	50.74	16.58	8.17	55.41	56.86	2.25	2.00	96.93
e2-standard-32	57.58	33.15	14.06	60.60	57.52	2.25	2.13	96.75

Enabling MSFS involves transitioning to a less computationally demanding virtual machine type whenever feasible and scaling up as required. As a result, the average daily operating cost of a virtual machine in USD is significantly reduced with MSFS – *Daily USD cost (with MSFS)*. For instance, the daily cost of an e2-standard-16 machine with static reservation – *Daily USD cost (without MSFS)* – would be USD 16.58 (specific to the europe-central2 location). However, by adjusting such an initial reference type within the e2-standard flavor to forecasts, the average cost can be reduced to approximately USD 8.17.

Qualitative indicators such as average CPU and RAM usage enable the assessment of resources that remain unused despite the application of MSFS. While the evaluation in this study relies solely on predefined machine types, utilizing custom machine types with finer vCPU and RAM level granularity can further optimize resource utilization. This is particularly noticeable in the case of e2-standard-2 selected as the reference type. As it is the lowest-tier variant in the e2-standard flavor hierarchy, further down-scaling is not possible, resulting in *Percentage cost reduction (with MSFS)* equal to zero. However, excessive resource occupancy may impact service and application performance stability, so aiming for near-perfect waste reduction may not be a desirable direction. Additionally, crucial metrics encompass the frequency of scaling events – *Scaling events*, denoting the average number of transitions between machine types (upscaling and down-scaling) per single virtual machine during the test set period. Excessive scaling can induce instability and incur additional overhead. Moreover, *Violation events* denote situations where the actual usage surpassed the forecasted predictions, impacting service availability and SLAs.

Adjusted reservation plans result from incorporating domain knowledge through administrative rules, modifying initial plans to enhance the interpretability of MSFS. Table 10 presents a summary of adjusted reservation plans. The administrative rule used involved recommending a more robust machine type within the e2-standard flavor hierarchy if the forecasted demand exceeded 80% of the initially suggested machine's computational capacity, in terms of either CPU or RAM.

Such an administrative rule prioritizes enhancing overall cloud environment performance over achieving consistent cost reduction. Consequently, it leads to a significant decrease in the number of violation events and an increase in service availability, ensuring compliance with SLAs despite achieving a less significant increase in cost-effectiveness compared to the initial reservation plans. Thus, administrative rules not only enhance the interpretability of MSFS decisions but also enable the following choice: maximizing cost-effectiveness, maintaining high QoS or a tradeoff between these two. Analyzing both Tables 9 and 10 reveals that the greater the computational capacity of the virtual machine, the higher the probability of resource wastage, where versatile MSFS can yield particularly favorable results. As shown in Table 9, the highest average percentage cost reduction was achieved for the e2-standard- 32 reference type compared to initial reservation plans – 57.57%. However, we use the results obtained from the adjusted reservation plans as a reference point, since these embody the desired balance between resource utilization and performance. Consequently, with respect to adjusted reservation plans and Table 10, the greatest average percentage cost reduction was achieved with experimental scenarios assuming e2-standard-32 as the reference type – 44.71%. It is worth noting that creating reservation plans enables committed use discounts to be leveraged, which can further aid in cost optimization.

 doi:10.20944/preprints202409.0616.v1

Table 10. Averaged qualitative and quantitative indicators for adjusted reservation plans based on various reference virtual machine types.

Reference type	Percentage cost reduction (with MSFS)	Daily USD cost (without MSFS)	Daily USD cost (with MSFS)	Percentage CPU usage (with MSFS)	Percentage RAM usage (with MSFS)	Scaling events	Violation events	Percentage availability
e2-standard-2	0.00	2.07	2.07	17.69	22.13	0.00	0.00	100.00
e2-standard-4	20.87	4.14	3.28	29.52	36.42	0.75	0.00	100.00
e2-standard-8	34.02	8.29	5.47	35.33	40.23	1.50	0.00	100.00
e2-standard-16	36.98	16.58	10.45	35.06	41.12	4.75	0.25	99.62
e2-standard-32	44.71	33.15	18.33	36.07	39.75	4.75	0.25	99.62

4.5. Summary

The application of a multi-faceted GSFM concept in MSFS aims to construct group-specific forecasting models, thereby striking a compromise between the GFM and LFM concepts in a knowledge-based manner. STG allows for the optimization of multi-time series forecasting, making it scalable in the face of a growing number of virtual machines while enabling GSFM to achieve improved evaluation metrics compared to competing approaches.

In general, the concept-centric approach underscores the meticulous attention given to both pre-prediction and post- prediction phases. Therefore, within MSFS, alongside data- driven STG, various mechanisms are integrated, including ensemble-based anomaly detection using HEADA, dynamic resource reservation, and addressing prediction uncertainties. While administrative rules and adjusted plans serve to mitigate discrepancies, continuous monitoring remains paramount. Thus, MSFS components are FinOps-driven, playing a pivotal role in the domain of applied machine learning within the context of cloud environments.

5. Conclusions

In this research, we introduced STG as an effective method for optimizing multi-time series forecasting, addressing the need for scalable and reliable solutions to enhance resource utilization in rapidly evolving cloud environments. STG further served as the basis for the GSFM concept, representing a tradeoff between the two contrasting approaches – GFM and LFM. The multivariate time series data originating from these are characterized by high complexity, dynamism, and variations in both dimensionality and lengths across individual signals. Nevertheless, MSFS efficiently accommodates these properties. During evaluation, our approach demonstrated both qualitative and quantitative benefits, resulting not only in outperforming the GFM and LFM concepts but also in significant cost savings based on resource reservation plan estimations, with average operational expense reductions of up to 44.71%.

Throughout the study, we consistently advocate incorporating domain-level metrics, as they enhance interpretability in applied machine learning and facilitate multi-criteria decision-making. Additionally, we consider leveraging the FinOps context to be crucial, enabling the challenges associated with cloud computing to be overcome. While cloud computing environments are highly appealing, achieving high stability and performance while optimizing costs and reducing resource wastage inevitably entails a tradeoff. The imperative for such a balance is apparent in both the design of MSFS and the outcomes of its evaluation. Additionally, the need for stability in decisions and verdicts when relying on machine learning has also been demonstrated with respect to HEADA, where the concepts of hierarchy, ensemble approach, voting mechanisms and weighting have been integrated to achieve greater outlier detection certainty. While various cloud environments may have diverse operational purposes, the MSFS concept emerges as cloud-agnostic due to its

 doi:10.20944/preprints202409.0616.v1

scalability, cost-awareness, and a range of options enabling it to be tuned to specific use cases. This solution can be utilized in both small and large cloud environments, accommodating transitions between these states, allowing for the injection of domain knowledge, and thus justifying actions undertaken in terms of cloud resource management. Resource usage prediction greatly improves allocation efficiency over traditional methods. Conventional cloud resource management models, often labeled as pessimistic, reserve resources with a substantial buffer above the expected demand to accommodate potential spikes and ensure that SLAs are met. This approach frequently leads to over-provisioning, wasted resources and higher operational costs, and it also lacks a data-driven basis. In contrast, our method leverages machine learning for dynamic predictions, aligning resource reservations more closely with anticipated demand. This approach minimizes waste and enhances cost effectiveness by adjusting allocations based on forecasted needs, providing a more efficient resource management solution. Nevertheless, dynamic resource allocation based on

forecasts introduces risks associated with prediction inaccuracies, which are less pronounced in traditional models. To mitigate these risks, implementing robust monitoring and anti-discrepancy mechanisms is essential. These safeguards help balance resource wastage with effective utilization.

Despite the fact that MSFS addresses many of the key challenges related to cloud resource usage optimization, the introduction of both the *Embedding* and *Grouping* stages may introduce additional overhead. Consequently, MSFS requires the embedding model to be up-to-date, necessitating periodic retraining to ensure that time series similarity detection remains reliable. Careful retraining, monitoring, and concept drift detection strategies may be necessary to maintain the high quality and effectiveness of *Embedding*. Additionally, the *Grouping* stage requires that forecasting models for groups remain current, ensuring that reservation plans are based on predictions that accurately reflect the dynamic characteristics of resource usage. Therefore, timely re-evaluation of virtual machine assignments to groups is critical. In a production environment, the frequency of these updates would depend on the dynamics of the environment and would need to be empirically determined. On the other hand, overly frequent re-evaluation or determining group membership based on an excessively short period of resource usage could lead to MSFS instability. Furthermore, our solution proves effective even in small and medium- sized cloud environments, particularly when only limited data is available. However, it is important to highlight that where more data is available, forecasts generally become more accurate. With smaller datasets, there is a higher risk of inaccurate predictions, a factor that needs to be clearly acknowledged. Additionally, in our future research, we aim to focus on optimizing resource usage in data-scarce cloud environments. We plan to explore the use of continuous learning, generative-based data augmentation and genetic algorithms while addressing challenges like concept drift and data drift within a FinOps context.

CRediT Authorship Contribution Statement

Mateusz Smendowski: Methodology, Software, Investigation, Writing – original draft, **Piotr Nawrocki**: Supervision, Conceptualization, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that support the findings of this study are available from Polcom, but restrictions apply to the availability of these data, which were used under license for the current study, and thus are not publicly available. The data are, however, available from the authors upon reasonable request and with permission of Polcom.

Acknowledgments

 doi:10.20944/preprints202409.0616.v1

The research presented in this paper was supported by funds from the Polish Ministry of Science and Higher Education allocated to the AGH University of Krakow. The authors would like to thank Polcom for providing the data used in the tests.

References

1. Alqahtani, D., 2023. Leveraging sparse auto-encoding and dynamic learning rate for efficient cloud workloads prediction. IEEE Access .
2. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A., 2020. Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA. p. 3395–3404.
3. Bader, J., Lehmann, F., Thamsen, L., Leser, U., Kao, O., 2024. Lotaru: Locally predicting workflow task runtimes for resource management on heterogeneous infrastructures. Future Generation Computer Systems 150, 171–185.
4. Bandara, K., Bergmeir, C., Hewamalage, H., 2021a. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. IEEE Transactions on Neural Networks and Learning Systems 32, 1586–1599.
5. Bandara, K., Bergmeir, C., Smyl, S., 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. Expert systems with applications 140, 112896.
6. Bandara, K., Hewamalage, H., Liu, Y.H., Kang, Y., Bergmeir, C., 2021b. Improving the accuracy of global forecasting models using time series data augmentation. Pattern Recognition 120, 108148.
7. Barbado, A., Óscar Corcho, Benjamins, R., 2022. Rule extraction in unsupervised anomaly detection for model explainability: Application to oneclass svm. Expert Systems with Applications 189, 116100.
8. Barut, C., Yildirim, G., Tatar, Y., 2024a. An intelligent and interpretable rule-based metaheuristic approach to task scheduling in cloud systems. Knowledge-Based Systems 284, 111241.
9. Barut, C., Yildirim, G., Tatar, Y., 2024b. An intelligent and interpretable rule-based metaheuristic approach to task scheduling in cloud systems. Knowledge-Based Systems 284, 111241.
10. Behera, I., Sobhanayak, S., 2024. Task scheduling optimization in heterogeneous cloud computing environments: A hybrid ga-gwo approach. Journal of Parallel and Distributed Computing 183, 104766.
11. Cai, B., Yang, S., Gao, L., Xiang, Y., 2023. Hybrid variational autoencoder for time series forecasting. arXiv preprint arXiv:2303.07048 .
12. Cai, T.T., Ma, R., 2022. Theoretical foundations of t-sne for visualizing high-dimensional clustered data. Journal of Machine Learning Research 23, 1–54.
13. Chen, W.J., Yao, J.J., Shao, Y.H., 2023. Volatility forecasting using deep neural network with time-series feature embedding. Economic research-Ekonomska istraživanja 36, 1377–1401.
14. Daraghmeh, M., Agarwal, A., Manzano, R., Zaman, M., 2021. Time series forecasting using facebook prophet for cloud resource management, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6.
15. Dogani, J., Khunjush, F., Mahmoudi, M.R., Seydali, M., 2023. Multivariate workload and resource prediction in cloud computing using cnn and gru by attention mechanism. The Journal of Supercomputing 79, 3437–3470.
16. Froese, V., Jain, B., Rymar, M., Weller, M., 2023. Fast exact dynamic time warping on run-length encoded time series. Algorithmica 85, 492–508.
17. Gabhane, J.P., Pathak, S., Thakare, N.M., 2023. A novel hybrid multi-resource load balancing approach using ant colony optimization with tabu search for cloud computing. Innovations in Systems and Software Engineering 19, 81–90.
18. Optimizing multi-time series forecasting for enhanced cloud resource utilization

[1] Gagolewski, M., Bartoszuk, M., Cena, A., 2021. Are cluster validity measures (in) valid? Information Sciences 581, 620–636.

19. Geetha, P., Vivekanandan, S., Yogitha, R., Jeyalakshmi, M., 2024. Optimal load balancing in cloud: Introduction to hybrid optimization algorithm. Expert Systems with Applications 237, 121450.
20. Gupta, R., Saxena, D., Gupta, I., Makkar, A., Singh, A.K., 2022a. Quantum machine learning driven malicious user prediction for cloud network communications. IEEE Networking Letters 4, 174–178.
21. Gupta, R., Saxena, D., Gupta, I., Singh, A.K., 2022b. Differential and triphase adaptive learning-based privacy-preserving model for medical data in cloud environment. IEEE Networking Letters 4, 217– 221.
22. Hao, Y., Zhao, C., Li, Z., Si, B., Unger, H., 2024. A learning and evolution-based intelligence algorithm for multi-objective heterogeneous cloud scheduling optimization. Knowledge-Based Systems 286, 111366.
23. Hewamalage, H., Bergmeir, C., Bandara, K., 2022. Global models for time series forecasting: A simulation study. Pattern Recognition 124, 108441.

24. Jastrzebska, A., Nápoles, G., Salgueiro, Y., Vanhoof, K., 2022. Evaluating time series similarity using concept-based models. KnowledgeBased Systems 238, 107811.
25. Jeong, B., Baek, S., Park, S., Jeon, J., Jeong, Y.S., 2023. Stable and efficient resource management using deep neural network on cloud computing. Neurocomputing 521, 99–112.
26. Jiang, J., Liu, F., Ng, W.W., Tang, Q., Zhong, G., Tang, X., Wang, B., 2023. Aerf: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing. Computer Communications 200, 86– 94.
27. Kim, S., Chung, E., Kang, P., 2023. Feat: A general framework for feature-aware multivariate time-series representation learning. Knowledge-Based Systems 277, 110790.
28. Li, H., Zheng, W., Tang, F., Zhu, Y., Huang, J., 2023a. Few-shot time-series anomaly detection with unsupervised domain adaptation. Information Sciences 649, 119610.
29. Parekh, Ruchit, and Charles Smith. "Innovative AI-driven software for fire safety design: Implementation in vast open structure." World Journal of Advanced Engineering Technology and Sciences 12.2 (2024): 741-750.
30. fficient time series augmentation methods, in: 2020 13th International
31. Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1004–1009.
32. Liu, P., Liu, J., Wu, K., 2020b. Cnn-fcm: System modeling promotes stability of deep learning in time series prediction. Knowledge-Based Systems 203, 106081.
33. Liu, Z., Zhang, J., Li, Y., 2022. Towards better time series prediction with model-independent, low-dispersion clusters of contextual subsequence embeddings. Knowledge-Based Systems 235, 107641.
34. Mahadevan, A., Mathioudakis, M., 2024. Cost-aware retraining for machine learning. Knowledge-Based Systems 293, 111610.
35. Nalmpantis, C., Vrakas, D., 2019. Signal2vec: Time series embedding representation, in: International conference on engineering applications of neural networks, Springer. pp. 80–90.
36. Nawrocki, P., Smendowski, M., 2023. Long-term prediction of cloud resource usage in high-performance computing, in: International Conference on Computational Science, Springer. pp. 532–546.
37. Nawrocki, P., Smendowski, M., 2024a. Finops-driven optimization of cloud resource usage for high-performance computing using machine learning. Journal of Computational Science , 102292.
38. Nawrocki, P., Smendowski, M., 2024b. Optimization of the use of cloud computing resources using exploratory data analysis and machine learning. Journal of Artificial Intelligence and Soft Computing Research 14, 287–308.
39. Nawrocki, P., Sus, W., 2022. Anomaly detection in the context of longterm cloud resource usage planning. Knowledge and Information Systems 64, 2689–2711.
40. Osypanka, P., Nawrocki, P., 2023. Qos-aware cloud resource prediction for computing services. IEEE Transactions on Services Computing 16, 1346–1357.
41. Parekh, Ruchit, and Charles Smith. "Innovative AI-driven software for fire safety design: Implementation in vast open structure." World Journal of Advanced Engineering Technology and Sciences 12.2 (2024): 741-750.
42. D., Gupta, I., Gupta, R., Singh, A.K., Wen, X., 2023a. An ai-
43. driven vm threat prediction model for multi-risks analysis-based cloud cybersecurity. IEEE Transactions on Systems, Man, and Cybernetics: Systems .
44. Saxena, D., Gupta, R., Singh, A.K., Vasilakos, A.V., 2023b. Emerging vm threat prediction and dynamic workload estimation for secure resource management in industrial clouds. IEEE Transactions on Automation Science and Engineering .
45. Shu, W., Cai, K., Xiong, N.N., 2021. Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing. Future Generation Computer Systems 124, 12–20.
46. Si, W., Pan, L., Liu, S., 2022. A cost-driven online auto-scaling algorithm for web applications in cloud environments. KnowledgeBased Systems 244, 108523.
47. Singh, A.K., Gupta, R., 2022. A privacy-preserving model based on differential approach for sensitive data in cloud environment. Multimedia Tools and Applications 81, 33127–33150.
48. Singh, A.K., Swain, S.R., Saxena, D., Lee, C.N., 2023. A bio- inspired virtual machine placement toward sustainable cloud resource management. IEEE Systems Journal .
49. Song, K., Yu, Y., Zhang, T., Li, X., Lei, Z., He, H., Wang, Y., Gao, S.,
50. 2024. Short-term load forecasting based on ceemdan and dendritic deep learning. Knowledge-Based Systems 294, 111729.
51. Storment, J., Fuller, M., 2023. Cloud FinOps. O'Reilly Media, Inc.
52. Tran, M.N., Kim, Y., 2024. Optimized resource usage with hybrid auto-scaling system for knative serverless edge computing. Future Generation Computer Systems 152, 304–316.
53. Uribarri, G., Mindlin, G.B., 2022. Dynamical time series embeddings in recurrent neural networks. Chaos, Solitons & Fractals 154, 111612.

 doi:10.20944/preprints202409.0616.v1

54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.
55. Wang, Y., Long, H., Zheng, L., Shang, J., 2023. Graphformer: Adaptive graph correlation transformer for multivariate long sequence time series forecasting. Knowledge-Based Systems , 111321.
56. Wang, Z., Pei, C., Ma, M., Wang, X., Li, Z., Pei, D., Rajmohan, S., Zhang, D., Lin, Q., Zhang, H., et al., 2024. Revisiting vae for unsupervised time series anomaly detection: A frequency perspective, in: Proceedings of the ACM on Web Conference, pp. 3096–3105.
57. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L., 2022. Transformers in time series: A survey. arXiv preprint arXiv:2202.07125 .
58. Yadav, H., Thakkar, A., 2024. Noa-lstm: An efficient lstm cell archi- tecture for time series forecasting. Expert Systems with Applications 238, 122333.
59. Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., Niu, Z., 2024. Frequency-domain mlps are more effective learners in time series forecasting. Advances in Neural Information Processing Systems 36.
60. Yuan, H., Liao, S., 2024. A time series-based approach to elastic kubernetes scaling. Electronics 13.
61. Zhang, F., Guo, T., Wang, H., 2023. Dfnet: Decomposition fusion model for long sequence time-series forecasting. Knowledge-Based Systems 277, 110794.
62. Zhu, J., Bai, W., Zhao, J., Zuo, L., Zhou, T., Li, K., 2023. Variational mode decomposition and sample entropy optimization based trans- former framework for cloud resource load prediction. Knowledge- Based Systems 280, 111042.