

Forecasting virtual machine resource utilization in cloud computing: a hybrid artificial intelligence approach

Rim Doukha, Abderrahmane Ez-Zahout, Aristide Ndayikengurukiye

Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco

Article Info

Article history:

Received May 17, 2024

Revised Oct 1, 2024

Accepted Oct 7, 2024

Keywords:

ARIMA

Artificial intelligence

Cloud computing

Linear regression

LSTM

Machine learning

ABSTRACT

Cloud computing has transformed the management of IT infrastructures by providing scalable, flexible, and cost-effective solutions. However, efficient resource management in cloud environments remains a significant challenge, as over-provisioning or under-provisioning of resources can lead to unnecessary costs or degraded performance. Accurate forecasting of virtual machine (VM) resource utilization is crucial for optimizing resource allocation, reducing operational expenses, and ensuring compliance with service level agreements (SLAs). This study aims to address these challenges by developing a hybrid forecasting model that combines the strengths of auto regressive integrated moving average (ARIMA), linear regression (LR), and long short-term memory (LSTM) techniques. By integrating these methods, our model provides more accurate predictions and better adaptability to various workload patterns, helping cloud service providers and users to make informed decisions about resource allocation, ultimately reducing costs. The data was collected from multiple EC2 instances and processed using amazon web services (AWS) Glue with Spark. The experimental results demonstrate that the hybrid model outperforms individual models such as ARIMA, LR, and LSTM in terms of accuracy for forecasting memory, CPU, and disk utilization, offering a more effective solution for managing cloud resources efficiently.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Rim Doukha

Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University

Rabat, Morocco

Email: doukharim@gmail.com

1. INTRODUCTION

Cloud computing platforms are essential in today's IT infrastructure, providing vast resources and services, such as computing power, storage, and bandwidth, to customers on demand. These platforms offer scalability and dependability to ensure that quality of service (QoS) requirements outlined in service level agreements (SLA) are met, making use of virtualization to optimize resource allocation and efficiency [1]. Within these infrastructures, CPU capacity, memory, and disk utilization are indispensable for deploying any application in the cloud. Failure to manage and utilize these services efficiently can result in escalating costs [2]. Therefore, optimizing virtual machines (VMs) is paramount to ensure peak performance and optimal resource utilization [3]. This optimization is also crucial for cost-effectiveness [4].

In the realm of optimizing VMs in the cloud, two primary approaches emerge, human-driven optimization and predictive modeling with statistical optimization. The human-driven optimization relies on human expertise to interpret system behavior [5], analyze historical data, and make informed predictions about future workloads. This approach excels in understanding complex patterns and adapting VMs

configurations. This approach is particularly valuable when dealing with intricate, non-linear patterns and dynamic workloads that may be challenging for automated systems to interpret accurately.

On the other hand, predictive modeling and statistical optimization offer a more automated and proactive solution to VMs optimization. By leveraging advanced statistical methods and machine learning algorithms such as auto regressive integrated moving average (ARIMA), linear regression (LR), and long short-term memory (LSTM), these approaches enable users to forecast resource requirements and dynamically adjust VMs configurations. The automated nature of this method ensures quick responses to changing conditions, allowing for efficient resource allocation without constant human intervention. This approach becomes crucial in scenarios where workloads exhibit patterns that can be effectively captured and predicted through statistical modeling.

Typically, the accuracy and effectiveness of statistical methods are enhanced by combining several techniques [6]. For instance, ARIMA is adept at capturing temporal patterns, LR provides insights into linear relationships, and LSTM excels in modeling intricate sequences and correlations. Therefore, combining the features of these methods can result in a more accurate predictive model. By predicting VMs utilization, cloud providers can better utilize their infrastructures, ensuring that resources are allocated effectively to meet varying workload demands. This helps as well to reduce operational costs, and enhanced scalability.

Various research efforts have focused on predicting VM resource usage in cloud environments. For instance, Wu *et al.* [7] developed an adaptive hybrid prediction scheme based on auto-regression for forecasting CPU load in computational grids. They integrated adaptive parameters and the Savitzky-Golay filter to enhance performance. Hybrid models, merging ARIMA and ANN techniques, have shown superiority in forecasting CPU and memory usage.

Bi *et al.* [8], propose a hybrid deep learning model, BG-LSTM, which combines Bi-directional LSTM (Bi-LSTM) and grid LSTM for predicting workload and resource usage in cloud systems. Their approach effectively captures bi-directional dependencies and time-frequency domain features, leading to improved prediction accuracy. They applied a Savitzky-Golay filter to smooth the data and tested the model on Google Cluster traces. The results showed that the proposed method outperformed traditional models like ARIMA and other deep learning models, such as standard LSTM and SVM, particularly in long-term predictions.

Karim *et al.* [9], present BHyPreC, a hybrid recurrent neural network (RNN) model designed to predict CPU workload in cloud VMs. The model combines Bi-LSTM, LSTM, and gated recurrent unit (GRU) layers to enhance the prediction accuracy of non-linear time series data. The architecture includes a 1D convolutional neural network (CNN) layer for feature extraction and applies a sliding window approach to manage sequential data. Experimental results demonstrate that BHyPreC outperforms traditional methods like ARIMA and other RNN-based models, particularly in long-term predictions, offering a more accurate and stable forecasting tool for cloud resource management.

Banerjee *et al.* [10], propose a framework for improving resource utilization and energy efficiency in cloud data centers through a multi-step-ahead CPU workload prediction approach. The framework leverages machine learning techniques to predict the future CPU resource consumption of VMs based on historical data. A sliding window approach is employed to manage the time-series data, and a re-training mechanism is implemented to maintain prediction accuracy over time. The predicted CPU workload is then used to optimize the placement of VMs on physical machines (PMs), leading to a significant reduction in energy consumption and improved resource utilization compared to conventional methods. The approach is validated using real workload data from Bitbrains, demonstrating superior performance in maintaining data center efficiency.

Matoussi and Hamrouni [11] introduced a novel approach for predicting the influx of requests to a software as a service (SaaS) application, aiming to optimize resource allocation in cloud environments. Their method leverages the temporal locality principle and dynamically assigns weights to recent data points to forecast future demand accurately. The study addresses resource management challenges in the cloud by employing prediction techniques to anticipate service demand. It categorizes existing prediction approaches into four dimensions and highlights their key characteristics. By considering various factors such as service performance, workload, and resource utilization, the method adjusts the computation time using a sliding window and assesses the stability of recent workload.

Yavad and Yavad [12] introduce a predictive analysis utilizing deep learning techniques, specifically LSTM, to forecast future server loads in time series data. Their study employs this analysis to allocate the appropriate number of containers for running applications, effectively managing workload fluctuations. Additionally, their model implements a hybrid approach of vertical and horizontal elasticity to ensure efficient resource allocation and scalability.

Chen *et al.* [13] propose a resource usage prediction method called RPTCN, which utilizes temporal convolutional networks (TCNs), a form of deep learning, in cloud systems. They enhance TCNs by

incorporating a fully connected layer and an attention mechanism to enhance prediction accuracy. To understand the relationship between resource usage across different dimensions over time, the authors conduct correlation analysis to identify relevant performance indicators as multidimensional feature inputs for prediction. Their evaluation demonstrates significant improvements in prediction accuracy compared to baseline methods.

Devi and Valli [14] focused on resource management within infrastructure as a service (IaaS) environments. It emphasizes the importance of effective resource management to meet demand as it arises. Specifically, the study explores the role of predicting CPU and memory utilization in cloud resource provisioning. To achieve this, the study proposed a hybrid ARIMA-ANN model for forecasting future CPU and memory utilization. The hybrid model combines linear components detected by the ARIMA model with nonlinear components identified by the artificial neural network. By leveraging the residuals derived from the ARIMA model, the ANN enhances the recognition and amplification of nonlinear patterns in CPU and memory utilization traces. This approach enables the prediction of resource utilization patterns by considering both linear and nonlinear components, providing valuable insights for resource management in cloud environments.

Vila *et al.* [15] explores the use of trend analysis and time series forecasting techniques to enhance VM consolidation in cloud datacenters. By predicting near-future trends in VM resource usage and host availability, the study aims to improve scheduler decisions regarding VM migration and host allocation. Results indicate significant reductions in migrations, SLA violations, and energy consumption.

Ullah *et al.* [16] proposed an intelligent computing framework for predicting cloud VM resources in multivariate time-series data. The framework utilizes Bi-LSTM forecasting, considering multiple resource parameters such as CPU usage, memory usage, CPU cores, disk throughput, and network throughput. By leveraging deep learning techniques, the study aims to improve the accuracy of workload prediction and resource management in cloud data centers. Evaluation using real workload traces demonstrates the effectiveness of the proposed framework in accurately forecasting cloud resources, potentially enhancing resource allocation efficiency in cloud environments.

In the context of VM resource utilization prediction, existing research debates the relative effectiveness of linear versus nonlinear methods [17]. While some studies suggest that statistical and linear models outperform ANNs, particularly when strong correlations exist between independent variables, others argue that ANNs are more effective in capturing complex, nonlinear relationships. Given these contrasting perspectives, hybrid models have gained attention in time series forecasting, as they combine the strengths of multiple approaches, thereby minimizing the risk of errors associated with relying on a single method.

Building on this foundation, our study proposes a novel hybrid forecasting model designed to predict CPU, memory, and disk utilization in cloud computing environments. Unlike previous models that primarily focus on a single resource type, our hybrid approach integrates the capabilities of ARIMA for temporal pattern recognition, LR for identifying linear relationships, and LSTM for modeling complex sequences and dependencies. This integration is intended to provide a more holistic and accurate solution for resource management in cloud environments, adaptable to varying workload patterns.

The proposed hybrid model is designed with adaptability, making it applicable across a wide range of cloud infrastructures. This adaptability not only optimizes resource allocation but also lays the groundwork for future applications in VM placement strategies, an area critical for enhancing cost-efficiency and overall performance in cloud data centers. Table 1 (in Appendix) provides a summary of the main works presented, offering a comparative overview of the methods and their respective contributions to the field. The rest of this paper is organized as follows: section 2 described the method used in this study. Results and discussion are presented in section 3 and section 4 concludes this work and provides future directions.

2. METHOD

In this study, we develop a comprehensive hybrid model to predict CPU, memory, and disk utilization in cloud environments. The following section outlines the specific steps and techniques employed in developing and validating the predictive model, starting with data preprocessing and moving through to model implementation and evaluation. Our approach is designed to provide a robust and adaptable solution for efficient resource management across diverse cloud platforms.

2.1. High level architecture for predicting VM resources

To establish a robust methodology for forecasting VM resource utilization, including CPU, memory consumption and disk usage, we begin by collecting performance metrics from a range of VM using amazon web services (AWS). In this setup, three distinct VMs are instantiated, each running different workloads. The resulting metrics are stored in CSV format within amazon simple storage service (S3), a highly scalable storage solution known for its durability, availability, security, and performance [18].

The overall architecture used to collect, process, and analyze the data is illustrated in Figure 1. This figure provides a visual overview of the process flow, from data collection to the application of the hybrid model for forecasting resource utilization. Subsequently, Spark AWS Glue with spark was used to execute essential processing tasks on the accumulated data to ensure its quality and consistency. AWS Glue offers a serverless data integration solution, empowering analytics users to seamlessly discover, prepare, and integrate data from diverse sources. This service supports a spectrum of functionalities, including analytics, machine learning, and application development. AWS Glue enables connectivity to over 70 data source and facilitates centralized data management through a unified catalog.

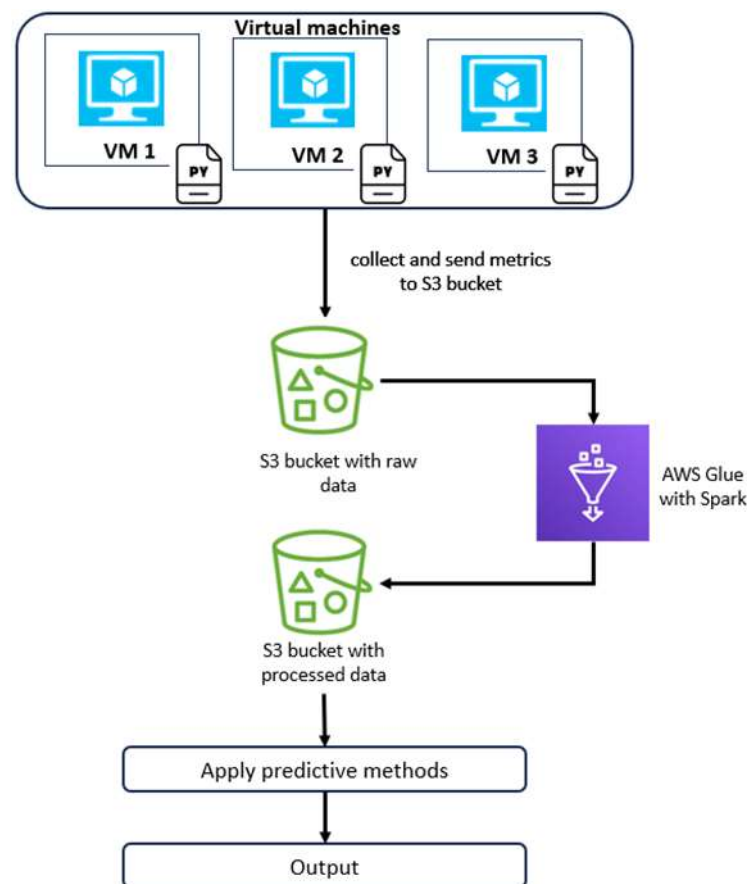


Figure 1. Architecture for data collection, processing, and resource utilization forecasting

Following the processing phase, the refined data was stored in a dedicated S3 bucket specifically designated for processed data. At this preparatory stage, the architecture employs our proposed hybrid model on the processed data to forecast future resource demands for VMs. The model combines LR simplicity, ARIMA's temporal insights, and LSTM's ability to learn intricate dependencies, resulting in more precise and robust optimization. The specific metrics used in this study, which are critical for understanding resource utilization across different workloads, are presented in Table 2. The following subsection goes into details about LR, ARIMA, and LSTM, explaining their individual strengths and advantages in the context of VM resource forecasting. Additionally, it outlines the benefits offered by the hybrid model, highlighting how the combination of these techniques improves predictive accuracy and adaptability.

2.2. Hybrid model description

The hybrid model used for forecasting VMs metrics merges three statistical and machine learning models. In this part, we will study the advantages and limitations of each of the models used separately and discuss the strengths of the proposed hybrid model.

Table 2. VMs metrics

| Metric | Definition |
|---------------------------|--|
| Timestamp | Time when the data was recorded |
| CPU percent 1 to 8 | CPU percentage usage for each of the eight cores |
| CPU stats ctx switches | Number of context switches in the CPU |
| CPU stats interrupts | Number of interrupts handled by the CPU |
| CPU stats soft interrupts | Number of software interrupts handled by the CPU |
| CPU stats syscalls | Number of system calls made by the CPU |
| CPU frequency curent | Current frequency of the CPU |
| CPU used percentage | Overall CPU usage percentage |
| CPU count | Number of CPUs in the system |
| Virtual memory total | Total virtual memory in the system |
| Virtual memory used | Virtual memory used |
| Virtual memory free | Free virtual memory |
| Virtual memory percent | Percentage of virtual memory used |
| Swap memory total | Total swap memory available |
| Swap memory used | Swap memory used |
| Swap memory free | Free swap memory |
| Swap memory percent | Percentage of swap memory used |
| Swap memory sin | Swap memory in (from disk) |
| Swap memory sout | Swap memory out (to disk) |
| Disk usage total | Total disk space |
| Disk usage used | Used disk space |
| Disk usage free | Free disk space |
| Disk usage percent | Percentage of disk space used |
| Disk io read count | Number of read operations on the disk |
| Disk io write count | Number of write operations on the disk |
| Disk io read bytes | Number of bytes read from the disk |
| Disk io write bytes | Number of bytes written to the disk |
| Disk io read time | Time spent on disk reads (in milliseconds) |
| Disk io write time | Time spent on disk writes (in milliseconds) |
| VM name | Name of the virtual machine or server where the data was collected |

2.2.1. Linear regression

LR is a statistical technique employed for modelling the relationship between one or more independent variables and a dependent variable [19]. At its core, LR seeks to establish a linear equation that best represents the relationship between input features and the target variable [20]. This equation takes the form:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

Where:

y represents the dependent variable (e.g., CPU utilization or memory usage)

X₁, X₂, ..., X_n denote the independent variables (e.g., CPU frequency)

β₀, β₁, β₂, ..., β_n are the coefficients to be estimated

This linear relationship facilitates prediction by allowing us to estimate the value of the dependent variable (e.g., CPU utilization) based on the known values of the independent variables (e.g., CPU frequency). By analyzing historical data and fitting a LR model, we can extrapolate future resource requirements for VMs. LR simplicity and interpretability make it a valuable tool for trend analysis, enabling us to identify and quantify the impact of each predictor variable on the target variable. Furthermore, its computational efficiency, especially for large datasets, enables swift implementation and training, making it suitable for real-time forecasting tasks in cloud environments [21].

However, it’s crucial to acknowledge the assumptions and limitations of LR. The method relies on the assumption of linearity between the independent and dependent variables, and its performance may be compromised in the presence of outliers or highly correlated predictors. Additionally, LR struggles with capturing non-linear relationships and is susceptible to multicollinearity issues when dealing with highly correlated predictors.

2.2.2. ARIMA

ARIMA, an acronym for auto regressive integrated moving average, is a well-known time series forecasting technique that integrates auto regression (AR), integrated (I), and moving averages (MA) [22]. In an ARIMA model denoted as ARIMA (p, d, q), “p” represents the order of the auto regressive component, “d” signifies the degree of differencing, and “q” denotes the order of the moving average component [23].

Here’s a breakdown of each component of ARIMA model:

- The AR part captures the relationship between an observation and a number of lagged observations (i.e., past values);
- The integrated (I) part involves differencing the raw observations to make the time series stationary;
- The MA part captures the relationship between an observation and a residual error from a moving average model applied to lagged observations.

While ARIMA is renowned of simplicity and versatility, it operates under the assumption of linearity in the relationship between past and future values [24]. Additionally, it may struggle with capturing long-term dependencies and nonlinear trends in the data. Nevertheless, ARIMA remains a valuable tool for time series forecasting, especially when dealing with stationary data and short-term patterns.

2.2.3. LSTM

LSTM, an acronym for long-short term memory networks, represents a sophisticated class of RNN architecture designed to address the challenges associated with modelling sequential data [25], particularly the vanishing gradient problem inherent in traditional RNNs. LSTMs excel in capturing intricate, long-term dependencies in sequences, making them well-suited for tasks such as time series forecasting and natural language processing [26].

The Figure 2 illustrates LSTM architecture. It consists of several components, including input gates, forget gates, output gates, and a memory cell. Each component plays a crucial role in processing and retaining information over extended sequences.

- Input gate: this gate determines which information from the current input should be stored in the memory cell. It regulates the flow of information based on the input data and the current state.
- Forget gate: it decides which information from the previous state of the memory cell should be discarded or forgotten. It selectively updates the memory cell by removing irrelevant information.
- Output gate: the output gate controls the flow of information from the memory cell to the output of the LSTM unit. It determines which information should be passed on to the next layer of the network or used for making predictions.
- Memory cell: the memory cell stores and maintains information over time. It retains important information from past time steps and updates its state based on input data and gate operations.

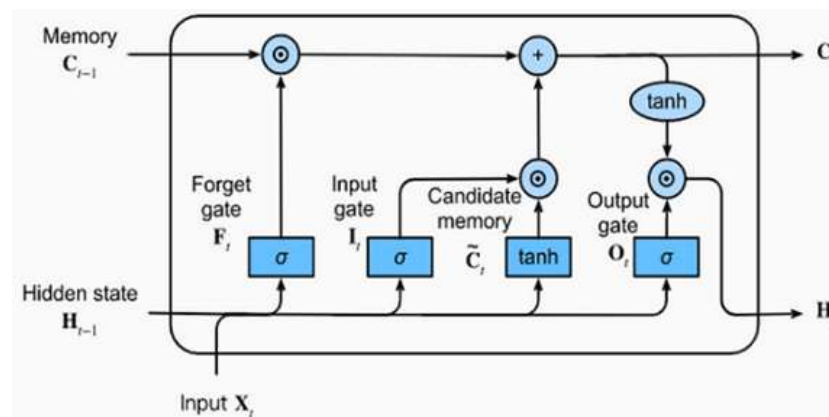


Figure 2. LSTM architecture

The LSTM architecture allows for the effective capture of long-term dependencies in sequential data by selectively retaining and updating information through the memory cell and gate mechanisms. This capability makes LSTM well-suited for tasks such as time series forecasting and natural language processing, where understanding contextual relationships over extended sequences is crucial for accurate predictions.

Despite their effectiveness, LSTMs come with computational demands due to their complex structure. Achieving optimal performance often requires a substantial amount of training data. Additionally, tuning the numerous hyperparameters of an LSTM can be a non-trivial task. While LSTMs offer unparalleled capabilities in capturing temporal dependencies, users should carefully consider computational resources, data availability, and tuning efforts when incorporating LSTMs into their modelling workflows.

After analyzing the advantages and limitations of each of the above models, we created a hybrid model that integrates ARIMA, LR, and LSTM to combine the strengths of each method and create a robust forecasting framework. In this hybrid approach, each model independently processes the same input data, generating its own set of predictions for CPU, memory, and disk utilization. The outputs of these models are then integrated through a weighted averaging method, where the final prediction is computed as a weighted sum of the individual model predictions. The weights assigned to each model's output were determined based on their performance during a validation phase, where each model was trained on 80% of the collected data and validated on the remaining 20%. The accuracy of each model was evaluated using the mean absolute percentage error (MAPE), with weights being inversely proportional to the MAPE scores, models with lower MAPE scores, indicating higher accuracy, were given greater weight. These weights were then normalized to ensure that their sum equals one, thereby maintaining a balanced contribution from each model. Additionally, the hybrid model incorporates a dynamic weight adjustment mechanism, allowing the weights to be periodically updated based on ongoing performance on new data. This approach ensures that the hybrid model remains adaptable and continues to deliver accurate predictions across varying workload patterns, thus enhancing the overall robustness of the forecasting framework.

To evaluate the performance of the proposed model, we trained our collected data using LR, ARIMA, LSTM, and the proposed hybrid model. We compared the results by calculating the accuracy of each model using MAPE. The next section describes the obtained results.

3. RESULTS AND DISCUSSION

After collecting data from EC2 instances and processing it, we created two separate datasets: one for training and one for testing the accuracy of the LR, ARIMA, LSTM, and hybrid models. The primary goal was to evaluate how well each model could predict VM resource utilization, specifically focusing on CPU, memory, and disk usage. Figure 3 compares the actual data with the predictions generated by each model, focusing on selected extracts of CPU, memory, and disk usage data. Figure 3(a) illustrates the CPU usage estimation, Figure 3(b) shows the memory usage estimation, and Figure 3(c) presents the disk usage estimation. Figures 4-6 further break down the results by illustrating the accuracy of each model in forecasting memory, CPU, and disk utilization, respectively.

Figure 3 shows that the hybrid model consistently provides predictions that are closer to the actual data compared to the individual models. This finding confirms the hypothesis that integrating different models can lead to superior performance, aligning with the work of Yadav and Yadav [12], who also demonstrated the benefits of hybrid approaches in workload prediction. However, our study extends this by not only focusing on CPU workload but also integrating memory and disk utilization predictions, providing a more comprehensive solution for cloud resource management.

For memory utilization, the hybrid model achieves an impressive accuracy of 98.99%, significantly outperforming LSTM (95.00%), ARIMA (88.50%), and LR (96.40%). This high level of accuracy indicates that the hybrid model effectively captures the complex patterns and dependencies inherent in memory utilization data. While Wu *et al.* [7] emphasized the advantages of adaptive hybrid models in grid computing, our results highlight the added value of incorporating LR alongside ARIMA and LSTM, particularly in cloud environments where diverse workload patterns are present.

Similarly, in the context of disk utilization, the hybrid model excels with an accuracy of 99.53%, surpassing LSTM (94.50%), ARIMA (92.50%), and LR (93.10%). The robustness of the hybrid approach in forecasting disk utilization highlights its ability to integrate and enhance the strengths of individual models, making it particularly effective for data with both long-term dependencies and short-term fluctuations. This result builds on the findings of Banerjee *et al.* [10], who focused on CPU workload prediction, by demonstrating that our hybrid approach is versatile enough to handle multiple resource types.

However, in predicting CPU utilization, the hybrid model achieves an accuracy of 89.50%. While this is the best performance among the models tested, it is notably lower than the accuracies achieved for memory and disk utilization. This discrepancy could be due to the inherently complex and volatile nature of CPU usage patterns, which might involve sudden spikes and drops that are more challenging to predict. This finding suggests that further refinement in the model, such as incorporating additional features or fine-tuning the existing parameters, might be necessary to better capture the nuances of CPU usage. This aligns with the observations of Chen *et al.* [13], who highlighted the challenges of predicting highly dynamic workloads in cloud environments.

In comparison to the work by Matoussi and Hamrouni [11], who introduced a temporal locality-based approach for predicting SaaS request influx, our hybrid model provides a broader application by focusing on the prediction of CPU, memory, and disk utilization across different types of workloads. This adds significant value to our approach, as it offers a more holistic solution to resource management in cloud environments.

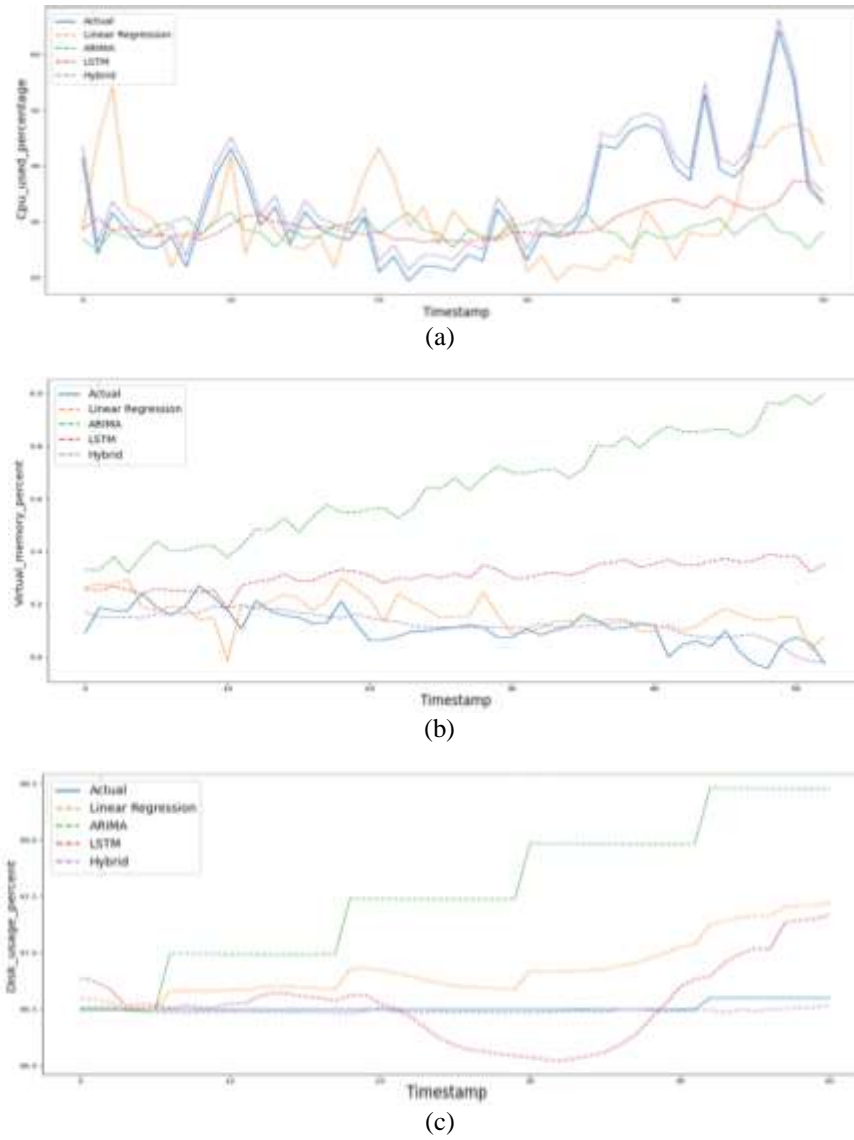


Figure 3. Compares the actual data (a) extract of CPU usage estimation, (b) memory usage estimation, and (c) disk usage estimation

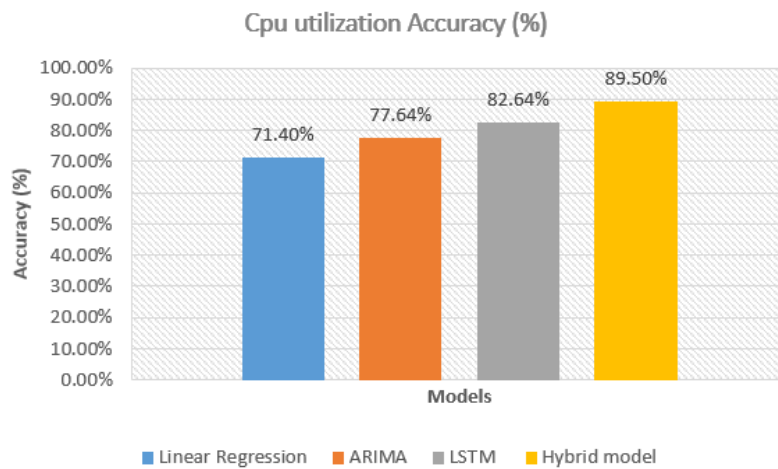


Figure 4. CPU usage accuracy

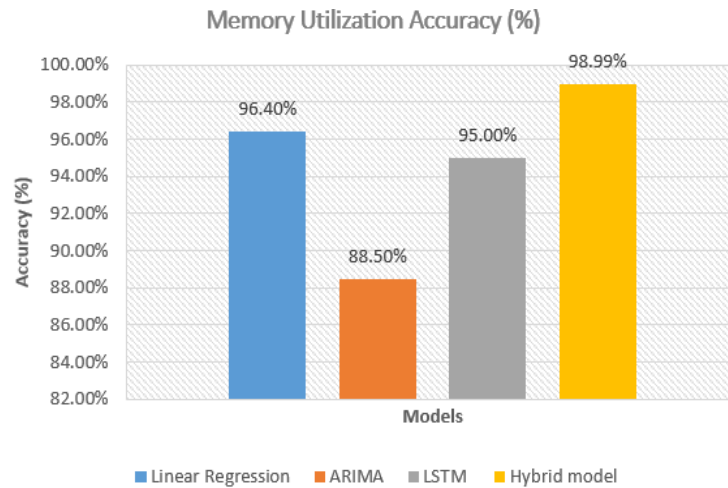


Figure 5. Memory usage accuracy

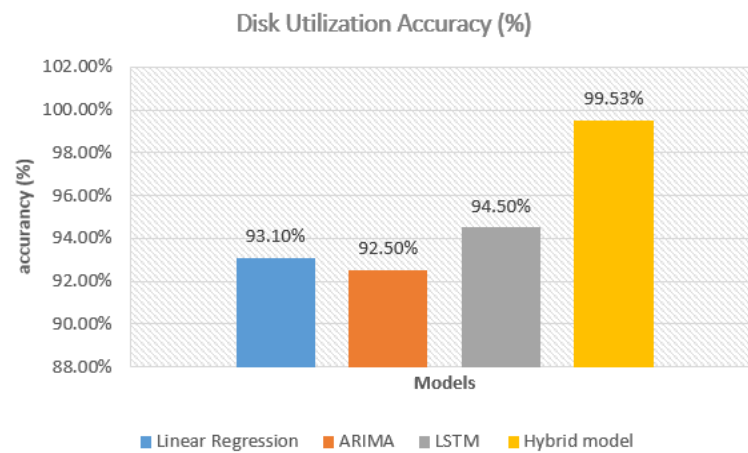


Figure 6. Disk usage accuracy

Our findings are consistent with earlier research, including the work of Bi *et al.* [8], which also demonstrated that hybrid models can substantially enhance prediction accuracy compared to single-model methods. However, our study advances the field by integrating LR, which is often overlooked in favor of more complex models like LSTM. The inclusion of LR helps capture linear trends that might not be fully addressed by LSTM or ARIMA alone. This combination is particularly effective in environments with a mix of linear and non-linear patterns, as demonstrated by our results.

Moreover, while Devi and Valli [14] focused on the integration of ARIMA and ANN for CPU and memory utilization, our work extends the hybrid approach by incorporating disk utilization, providing a more holistic solution. This is a significant added value, especially in cloud environments where multiple resources must be managed simultaneously. One limitation of our study is the relatively lower accuracy in predicting CPU utilization, which could be attributed to the complexity of CPU usage patterns. This limitation suggests that future work could explore additional modelling techniques or incorporate more granular data to improve predictions in this area.

Overall, our results underscore the effectiveness of the hybrid model in forecasting VM resource utilization across different metrics. The findings from this study are significant for cloud service providers, as they offer a accurate and adaptable solution for managing resources and optimizing VM placement. Future research could focus on enhancing the hybrid model’s performance in predicting CPU utilization by exploring additional features or alternative modelling techniques. Additionally, applying this model in real-time cloud environments and extending its capabilities to include other forms of resource prediction, such as network bandwidth and I/O operations, could further validate its effectiveness and generalizability.

4. CONCLUSION

In this study, we introduced a hybrid forecasting model that merges ARIMA, LR, and LSTM techniques to predict VM resource usage in cloud computing environments. By combining these methods, our model achieves superior forecasting accuracy and adaptability across various time patterns, effectively capturing both short-term changes and long-term trends in CPU, memory, and disk usage. Our experiments demonstrate that the hybrid model outperforms individual models, providing more accurate and reliable predictions. This is particularly important for cloud service providers who need precise forecasts to optimize resource allocation, reduce operational costs, and ensure service quality. The model's ability to combine the interpretability of ARIMA and LR with the deep learning capabilities of LSTM offers a powerful tool for understanding and managing complex resource utilization patterns.

These findings have significant implications for the field of cloud computing, particularly in enhancing the efficiency and sustainability of cloud infrastructures. The model's adaptability to various workload patterns also makes it a valuable asset for developing advanced VM placement strategies, contributing to the overall improvement of cloud service performance. Looking forward, future work will focus on refining and optimizing the hybrid forecasting model by exploring alternative architectures, fine-tuning hyperparameters, and incorporating additional features or data sources. We also plan to extend our predictions to include energy consumption, addressing a critical challenge in cloud computing. By doing so, we aim to contribute to the development of more energy-efficient cloud environments, which is increasingly important in the context of sustainable computing.

APPENDIX





Table 1. Comparative analysis of predictive models for VM resource utilization in cloud computing

| Article | Preprocessing | Prediction method | Prediction type | Workload/Platform | Key contributions/findings |
|------------------------------|----------------|--|-----------------------------------|--------------------------------------|---|
| Devi and Valli [14] | Yes | Hybrid ARIMA-ANN model | CPU and memory utilization | Google cluster, BitBrain datasets | Combines ARIMA and ANN to predict CPU and memory usage with improved accuracy. |
| Calheiros <i>et al.</i> [17] | None specified | ARIMA model | SaaS request workload | Cloud applications (SaaS) | Utilizes ARIMA for proactive resource allocation to enhance QoS. |
| Mbelli [8] | Yes | Deep learning (BG-LSTM) | CPU and memory utilization | Google cluster trace | Introduces a bi-directional LSTM model with grid configuration for better prediction in complex workload scenarios. |
| Wu <i>et al.</i> [7] | Yes | Adaptive hybrid model with ARIMA and ANN | CPU utilization | Grid computing (Grid5000, AuverGrid) | improve accuracy in grid workload forecasting. |
| Karim <i>et al.</i> [9] | Yes | Hybrid Bi-LSTM + LSTM + GRU | CPU utilization | Cloud virtual machines | Introduces a hybrid RNN model combining Bi-LSTM, LSTM, and GRU layers, with superior accuracy in non-linear time series data prediction. |
| Banerjee <i>et al.</i> [10] | None specified | LSTM, GRUED, GB, SVM | CPU and memory utilization | Cloud data centers | Proposes a framework for multi-step-ahead CPU workload prediction using ML models, improving resource utilization and energy efficiency in cloud data centers. |
| Yavad and Yadav [12] | Yes | LSTM | Network traffic | Cloud servers | Demonstrates the effectiveness of LSTM in predicting network traffic, improving resource utilization and response time in cloud environments. |
| Matoussi and Hamrouni [11] | Yes | Temporal locality-based prediction | SaaS request | Wikipedia, Bitbrains datasets | Proposes a method using temporal locality and dynamic sliding windows to predict the number of SaaS service requests, optimizing response time and resource allocation. |
| Vila <i>et al.</i> [15] | None specified | Combined prediction model (grey model and ARIMA) | CPU utilization | CloudSim simulation platform | Proposes a VM consolidation framework using a combined prediction model and energy-aware policies to reduce energy consumption, VM migrations, and SLA violations. |
| Our study | Yes | Hybrid AI model combining ARIMA, linear regression, and LSTM | CPU, memory, and disk utilization | AWS EC2 instances | Integrates multiple models to capture linear and non-linear patterns, achieving superior accuracy in VM resource forecasting (CPU, memory, and disk utilization). |





REFERENCES

- [1] A. Sunyaev, "Cloud computing," in *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, A. Sunyaev, Ed., Cham: Springer International Publishing, 2020, pp. 195–236. doi: 10.1007/978-3-030-34957-87.
- [2] R. Doukha, S. A. Mahmoudi, M. Zbakh, and P. Manneback, "A comparative analysis of convolution neural network models on amazon cloud service," in *AIIPCC 2022; The Third International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, Jun. 2022, pp. 1–7. Accessed: Apr. 22, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10025899>.
- [3] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, "Optimizing virtual machines placement in a heterogeneous cloud data center system," *IJCNA*, vol. 11, no. 1, p. 1, Feb. 2024, doi: 10.22247/ijcna/2024/224431.
- [4] A. Ndayikengurukiye, A. Ez-zahout, A. Aboubakr, Y. Charkaoui, and O. Fouzia, "Resource optimisation in cloud computing: comparative study of algorithms applied to recommendations in a big data analysis architecture," *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 65–75, 2021, doi: 10.14313/JAMRIS/4-2021/28.
- [5] B. Lin, "Human-driven optimization," PhD diss., Northwestern University, 2007, doi: 10.13140/2.1.3305.8882.
- [6] C. Kuranga, T. S. Muwani, and N. Ranganai, "A multi-population particle swarm optimization-based time series predictive technique," *Expert Systems with Applications*, vol. 233, p. 120935, Dec. 2023, doi: 10.1016/j.eswa.2023.120935.
- [7] Y. Wu, K. Hwang, Y. Yuan, and W. Zheng, "Adaptive workload prediction of grid performance in confidence windows," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 925–938, Jul. 2010, doi: 10.1109/TPDS.2009.137.
- [8] J. Bi, S. Li, H. Yuan, and M. Zhou, "Integrated deep learning method for workload and resource prediction in cloud systems," *Neurocomputing*, vol. 424, pp. 35–48, Feb. 2021, doi: 10.1016/j.neucom.2020.11.011.
- [9] Md. E. Karim, M. M. S. Maswood, S. Das, and A. G. Alharbi, "BHyPreC: a novel Bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine," *IEEE Access*, vol. 9, pp. 131476–131495, 2021, doi: 10.1109/ACCESS.2021.3113714.
- [10] S. Banerjee, S. Roy, and S. Khatua, "Efficient resource utilization using multi-step-ahead workload prediction technique in cloud," *Journal Supercomput*, vol. 77, no. 9, pp. 10636–10663, Sep. 2021, doi: 10.1007/s11227-021-03701-y.
- [11] W. Matoussi and T. Hamrouni, "A new temporal locality-based workload prediction approach for SaaS services in a cloud environment," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 3973–3987, Jul. 2022, doi: 10.1016/j.jksuci.2021.04.008.
- [12] M. Yadav and D. Yadav, "Workload prediction for cloud resource provisioning using time series data," in *Soft Computing for Problem Solving: Proceedings of SocProS*, 2021, pp. 447–459. doi: 10.1007/978-981-16-2712-5_37.
- [13] W. Chen, C. Lu, K. Ye, Y. Wang, and C.-Z. Xu, "RPTCN: resource prediction for high-dynamic workloads in clouds based on deep learning," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2021, pp. 59–69, doi: 10.1109/Cluster48925.2021.00038.
- [14] L. Devi and S. Valli, "Time series-based workload prediction using the statistical hybrid model for the cloud environment," *Computing*, vol. 105, pp. 1–22, Nov. 2022, doi: 10.1007/s00607-022-01129-7.
- [15] S. Vila, F. Guirado, and J. L. L rida, "Cloud computing virtual machine consolidation based on stock trading forecast techniques," *Future Generation Computer Systems*, vol. 145, pp. 321–336, Aug. 2023, doi: 10.1016/j.future.2023.03.018.
- [16] F. Ullah, M. Bilal, and S.-K. Yoon, "Intelligent time-series forecasting framework for non-linear dynamic workload and resource prediction in cloud," *Computer Networks*, vol. 225, p. 109653, Apr. 2023, doi: 10.1016/j.comnet.2023.109653.
- [17] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 449–458, Oct. 2015, doi: 10.1109/TCC.2014.2350475.
- [18] T. M. Mbelli, "Computational secure ORAM (COMP SE-ORAM) with $[\omega](\log n)$ Overhead: Amazon S3 case study – random access location," in *2019 IEEE Cloud Summit*, Washington, USA: Aug. 2019, pp. 99–102, doi: 10.1109/CloudSummit47114.2019.00022.
- [19] A. Tandon, A. Awasthi, and K. Pattanayak, *Comparison of different Machine Learning methods on Precipitation dataset for Uttarakhand*. 2023, p. 6. doi: 10.1109/ICAHC59020.2023.10431402.
- [20] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, Art. no. 2, Dec. 2020, doi: 10.38094/jastt1457.
- [21] T. M. H. Hope, "Chapter 4 - Linear regression," in *Machine Learning*, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 67–81. doi: 10.1016/B978-0-12-815739-8.00004-3.
- [22] K. Dmytro, T. Sergii, and P. Andiy, "ARIMA forecast models for scheduling usage of resources in IT-infrastructure," in *2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv: IEEE, Sep. 2017, pp. 356–360, doi: 10.1109/STC-CSIT.2017.8098804.
- [23] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah, "A review on time series forecasting techniques for building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 902–924, Jul. 2017, doi: 10.1016/j.rser.2017.02.085.
- [24] H. Mo, "Comparative analysis of linear regression, polynomial regression, and ARIMA model for short-term stock price forecasting," *Advances in Economics, Management and Political Sciences*, vol. 49, pp. 166–175, Dec. 2023, doi: 10.54254/2754-1169/49/20230509.
- [25] A. Gozuoglu, O. Ozgonenel, and C. Gezezin, "CNN-LSTM based deep learning application on Jetson Nano: estimating electrical energy consumption for future smart homes," *Internet of Things*, vol. 26, p. 101148, Jul. 2024, doi: 10.1016/j.iot.2024.101148.
- [26] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, "A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks," *Future Internet*, vol. 15, no. 8, Art. no. 8, Aug. 2023, doi: 10.3390/fi15080255.





BIOGRAPHIES OF AUTHORS

Rim Doukha     currently pursuing a Ph.D. in computer science at the Department of Computer Science, Faculty of Science, Mohammed V University, her research centers on cloud computing and artificial intelligence. She holds a Master's degree in IoT and Mobile Services from ENSIAS. Her professional experience includes internships in Morocco and France, work as a cloud computing research engineer in Belgium on a European project, and her current role as a project officer specializing in cloud computing and technical standardization in Luxembourg. She can be contacted at email: doukharim@gmail.com.



Abderrahmane Ez-Zahout     is associate professor - Department of Computer Sciences, Faculty of Sciences/Mohammed V University in Rabat. Ph.D. in Computer Sciences obtained ENSIAS IT College in Rabat, and Master Degree since 2003. Author of numerous articles and research works in the big data and AI era applied to different topics, and holder of several international projects. A coordinator of the Master's program in Software Development and Decision Engineering at the Computer Sciences Department of the Faculty of Sciences of Rabat and also Faculty member at Al Akhawayn University in Ifrane Morocco." Abderrahmane Ez-Zahout is currently Associate member of the Computer Sciencess laboratory and active member in IPSS (intelligent processing and security of systems) team research. Expert for the CMRPI resaech laboratoty (<https://www.cmrpi.ma/cmrpi-v2/>) in artificial intelligence and big data analytics and also expert in the TELEM, groupe ONET. He has worked extensively on intelligent systems and has served as an invited reviewer for several academic journals. In addition to his academic contributions, he is actively involved in NGOs, student associations, and the management of non-profit foundations. He can be contacted at email: abderrahmane.ezzahout@um5.ac.ma.



Aristide Ndayikengurukiye     he holds a Ph.D. in Cloud Computing and Artificial Intelligence from the Faculty of Science at Mohammed V University in Rabat, Morocco, and earned his M.Sc. in Computer Engineering from the University of Burundi in 2017. His research interests span cloud computing optimization, artificial intelligence, and big data. He can be contacted at email: lemure.dede@gmail.com.