Innovative Journal of Applied Science

Research Article

IJAS-24-002

Mathematical Models for Predicting Network Traffic in Cloud Computing Environments

Xingyu Liu[1*], Zhongkezhen Zhong[2], Chenxi Chen[3], Junjie Cha[4]

[1]*Department of Computer Science, Yanbian University, Jilin, China*

[2]*Department of Computer Science, Harbin Institute of Technology, Heilongjiang, China*

[3]*Department of Computer Science, University of Minnesota, Minneapolis, USA*

[4]*Department of Information and Communication, State Grid Jiangsu Electric Power Co., Ltd, Nanjing, China*

Corresponding Author: Xingyu Liu, Department of Computer Science, Yanbian University, Jilin, China, E-mail: studenthelper4@gmail.com

Received date: 28 September, 2024, **Accepted date:** 21 October, 2024, **Published date:** 01 November, 2024

Citation: Liu X, Zhong Z, Chen C, Cha J (2024) Mathematical Models for Predicting Network Traffic in Cloud Computing Environments. Innov J Appl Sci 1: 1.

Abstract

Effective network traffic prediction is crucial for optimizing resource allocation and ensuring efficient performance in cloud computing environments. In this research, mathematical models are developed to use historical data, factors, and other data to predict future network traffic patterns more effectively. In this context, we then compare an array of time series models such as ARIMA, LSTM, as well as the Prophet model so that we can determine the cloud environments most appropriate for each. These models include time and day of the week as well as the general activities of the users in the network in order to mimic the real flow of network traffics. The experimental results concern the efficiency of the proposed models as opposed to existing approaches and give a lot of information to network administrators and cloud service providers. The outcomes make a significant contribution as to the formulation of intelligent approaches in resource management and improve the dependability and performance of cloud computing environments.

Keywords: Network traffic prediction, Cloud computing, Mathematical modeling, Traffic forecasting algorithms, Network performance optimization

Introduction

Background and motivation

Cloud computing is one of the transformative technologies of the contemporary IT industry in that it provides efficient, flexible, and cheap solutions for the whole population. The ability of infrastructure, platforms, and software to facilitate the management of resources, applications, and data is possible through the use of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1]. As cloud services gain more ground in use within organizations, more pressure is placed on traffic within the cloud [2]. The patterns of cloud systems and operations depend on the implementation of the user's demands, the types of workloads, and the scenarios; therefore, the path of the network traffic must be optimized [3]. Coordinating this traffic is a rather delicate matter of resource allocation and usage of bandwidth and computing power [4]. Unfortunately, cloud networks are not devoid of various problems, such as fluctuations in traffic, lack of available bandwidth, and delays that negatively impact performance and quality of service [5]. Such problems underscore the importance of accurate forecasts of traffic in networks [6]. Forecasting is therefore employed in network planning as a tool by which service providers can avoid congestion and maintain the quality-of-service provision by scheduling the use of resources in relation to the traffic load expected in networks [7]. Traffic forecasting or accurate prediction has become one of the most important focuses of research with regard to cloud network performance [8]. Employing mathematical models may be the best way to solve this problem because such models can yield nearly accurate predictions based on past performance and computation formulas [9]. However, traditional approaches for forecasting network traffic are unsuitable for cloud computing systems mainly because of the complexity and dynamic workloads involved [10]. Thus, there is increasing interest in more complex, elastic, and adaptive methods for addressing the features of cloud network traffic analytics.

Problem statement

Traffic prediction in cloud computing networks is another fascinating task due to the dynamic and distributed structure of the network. The dynamics of workloads, flexibility of resources, and differences in the types of networks present in the network make traffic difficult to model and predict. Whenever the traffic predicted

in a network is erroneous and poor, traffic congestion and resource misallocation may occur in the network. These issues lead to increased costs of operation for cloud service providers and a decreased level of service for consumers.

Although there are a host of existing traffic prediction models in the relevant literature, most of them fail to capture the intricacies of contemporary cloud computing systems. Currently, there are challenges facing these models, mainly in regard to scalability and flexibility, particularly when managing changes in traffic patterns and achieving a good measure of precision while simultaneously considering computational complexity. However, the growth in the richness of cloud services and the continual diversification of cloud applications make it more challenging to develop models that can always predict traffic in real time. Solving these issues requires solutions that use enhanced mathematical formulations to improve the accuracy and efficiency of the models.

Objectives

This work seeks to propose, calibrate and verify mathematical formulas that can be used to forecast the network traffic of cloud computing facilities. The overall goal is therefore to develop new or enhanced models suitable for addressing the characteristics of cloud network traffic. There is also the aspects of variability and uncertainty inherent in such systems and designing new algorithms that can predict traffic loads in clouds. They prevent heavy traffic in the network, ensuring that the presented models are sufficiently abstract so that they can be reintroduced to extend the cloud network size with low computational complexity. The other is providing useful guidelines that will help CSPs apply these models for enhancing networks and resource efficiency in real-life situations.

Contributions

In this paper, the following original contributions to the field of cloud computing and network traffic management are presented. First, the idea of constructing a novel mathematical system that employs time series analysis techniques, stochastic models, and queuing theory is proposed to improve the forecasting of network traffic in cloud computing environments. This framework is aimed at coping with the features typical of cloud networks, such as fluxing and scalability. Second, to maintain the practicality of the study, real cloud network traffic data are used to assess the accuracy of the proposed models. This validation process also includes detailed performance evaluations under different circumstances, such as different loads on the network and different traffic types, to validate the use of these models under different conditions.

Literature Review

Overview of network traffic prediction in cloud computing

Network traffic prediction has remained one of the major subfields of network management research. Given that cloud computing has become so prevalent, obtaining rapid and accurate estimations of network traffic has become even more crucial [6]. Traffic patterns can be adequately predicted with the aim of smartly allocating resources to reduce latency and simultaneously provide efficient cloud services [11].

The first studies devoted to the prediction of network traffic were carried out in the context of traditional computer data networks, where traffic processes were much more stable and did not fluctuate as dynamically as in contemporary ‘cloud’ arrangements [5]. Most of these studies employed time series analysis, autoregressive models, and linear regression to forecast traffic volumes by using historical data. However, with the move to cloud computing, new issues such as traffic fluctuations, multitenancy, and dynamic resource allocation were encountered, issues that previous models were ill-equipped to solve [7]. To address these challenges, recent research has strived to develop complex models that incorporate the characteristics of cloud networks. Some of these are machine learning techniques, stochastic models, and integrated models, which use more than one method in a single model with the aim of enhancing the accuracy of the results [4]. Even if these theories have been developed, most of them still lack a number of characteristics, such as scalability, flexibility, and computational performance, especially in dynamic cloud environments [8].

Time series models for network traffic prediction

Time Series (TS) analysis of network traffic has been one of the most popular techniques for achieving the goal of network traffic prediction in general and in the context of traditional and cloud networks in particular [12]. ARIMA-type models, seasonal ARIMA models, and ETS-type models are used frequently because traffic data provide temporal dependency. These models are generally used when the number of vehicles entering the parking area follows a cyclic pattern that may be daily or weekly [6].

For instance, in cloud computing environments, time series models have been used to predict traffic at different time horizons on the scale of seconds, minutes, hours, etc. [10]. For instance, traffic prediction in virtualized networks, which experience sudden shifts in user behavior, can be performed using ARIMA to predict short-term traffic patterns [13]. Likewise, SARIMA has been used to forecast fluctuating patterns in traffic caused by periodic events or behaviors of users, while cloud traffic periodicity has been addressed by the model [5].

However, classic time series models can be problematic in cloud networks. Another issue that might be encountered in cloud environments due to the very high level of dynamic changes in traffic patterns is prediction inaccuracy [2]. In addition, time series models differ from other models used in data analytics and may need additional data to construct prediction models; however, these models are often insufficient for predicting the operation of new fast-growing cloud networks [3]. These drawbacks have led to the need to seek other models, such as machine learning and hybrid models, to improve the accuracy of cloud predictions [1].

Machine learning approaches to network traffic prediction

Network traffic prediction using machine learning techniques has attracted great focus in the recent past because of the high capability of learning highly complex patterns from data [1]. Compared to other statistical models that are more conventional, ML approaches can actually identify nonlinear relationships and interactions between variables; cloud networks are very dynamic, and the iron mixture can be very heterogeneous [14]. Most commonly, four models for traffic prediction are used, namely, Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and random forests [9]. ANNs are especially useful for network traffic prediction because of their versatility and ability to recreate complex traffic [6]. Several works with ANNs have been performed for traffic forecasting, traffic anomaly detection, and traffic type classification in cloud settings

[14]. For instance, other enhanced architectures of ANNs, including Long Short-Term Memory (LSTM), have been employed to enable the prediction of traffic data that have temporal migration, providing improved predictions when dependencies are temporal [12].

Support Vector Machines (SVMs) are another method for predicting gliomas and can result in classification and regression problems [12]. Cuong et al. utilized SVMs anticipating the traffic load in cloud data centers as a means of enhancing the provision of services in the domain of cloud computing [11]. The main strength of SVMs is their capacity when working with high-dimensional data and good performance when training sets are limited [5]. Network traffic prediction has also been performed using Random Forests (RF) and other methods under the umbrella of ensemble learning [14]. These methods use many decision trees to increase the accuracy of the prediction and avoid overtraining because, in clouds, traffic could be very unpredictable [13].

While the application of machine learning seems promising, these models are not without their limitations. A major concern is that to build these models, one requires large labeled datasets to feed the models into more than the first stage [9]. In most cloud platforms, gathering and annotating a sufficient amount of data can take considerable time and be very expensive [7]. Moreover, the training time of some of the models in the ML process can be long enough to be deployed in real-time prediction mode where prompt decisions/actions must be made [12].

Stochastic models in network traffic prediction

Stochastic models are based on the probability theory of expected traffic and hence differ from deterministic time series and machine learning models [2]. Such models are useful for architectures of cloud computing where traffic rates are unpredictable and arbitrary [6]. Stochastic models incorporate the probabilistic nature of network traffic, and in addition to the point values of the models, they give confidence bounds, which allow for measuring the spread of the predictions [8]. Thus, Markov models are among the most common stochastic techniques used in network traffic modeling [5]. All these models assume that the state of the network traffic at the subsequent moment depends on the current state and is not influenced by the series of events that have led up to it. This “memoryless” property makes modeling easier, and Markov models are more applicable in cloud environments where traffic may vary at a faster pace [10]. For instance, Markov models have been employed to forecast the state of traffic loads in cloud networks to plan for high-intensity traffic periods [4]. Another stochastic model is queuing theory, which has also been used in network traffic prediction, especially in relation to the traffic management of cloud resources [8]. Queuing models can estimate the rates of traffic arrival and waiting time for service to schedule the systems and expected bottlenecks in the network [6]. These types of models are useful when network traffic is variable; that is, they are useful for capturing queuing behavior, which causes congestion [7].

However, like in any other stochastic model, there are certain limitations associated with the use of the geometric Brownian model [14]. Work on these models has also shown that they can tend to be less accurate if the assumptions made on traffic patterns are inaccurate [6]. In the case of cloud environments, the traffic pattern is dynamic and may depend on several factors that contradict the assumptions made above [2]. Additionally, stochastic models can be complex in terms of computation, especially when applied to large-scale cloud networks with multiple traffic flows and considering multiple services.

Hybrid models and advanced techniques

Combined with the aspects of the time series approach, machine learning, together with stochastic techniques, has become a more suitable technique for predicting network traffic in cloud models [7]. These models are designed to exploit the advantages of each technique while minimizing shortfalls [10]. For instance, short-term dynamic variables may be captured via time-series analysis, while more complex and nonlinear relationships may be analyzed via machine learning algorithms [3]. Hybrid ARIMA-ANN models are a perfect example of this approach because they involve the use of both ARIMA models and ANN models [4]. The advantages of ARMA models include linear predictions, which are enriched with the nonlinear capabilities of ANNs in predicting patterns [5]. Another example of a hybrid model for predicting network traffic is the combination of ARIMA-SVM, where the ARIMA model is used for linear components of the traffic pattern, while the SVM model is used for nonlinear components [1].

Another hybrid model is a stochastic model with an Artificial Neural Network (ANN), which allows the integration of the uncertainty of the traffic in the cloud model to the architecture of an ANN [14]. Stochastic models such as Markov models or Poisson models have also been merged with deep learning techniques, such as Long Short-Term Memory (LSTM), to capture the temporal dynamics of cloud traffic and make accurate predictions [11]. However, these hybrid models also have challenges [6]. They are computationally intensive and require the knowledge and understanding of various methodologies, which makes their implementation more difficult, particularly compared to the implementation of single-method models.

Cloud Manufacturing Platform and Task Scheduling

In CMfg, there is a central cloud platform that plays the role of an intermediary between consumers and the providers of services. Clients provide work, while vendors supply virtualized capabilities. It breaks large tasks into small chunks of tasks, assigns them to the right resource, and then plans the schedule for performing these subtasks.

Mathematical Model for Task Scheduling

To optimize resource allocation and task execution, a mathematical model is proposed. The model considers the following factors:

- **Objective functions:** Time (T), Cost (C), Quality (Q), And Utilization (U).
- **Decision variables:** x_ij, indicating whether task i is assigned to resource j.
- **Constraints:** Time, cost, quality, and utilization limits.

Mathematical Formulation

Minimize: $\sum (a1* T + a2* C - a3* Q - a4* U)* x\_ij$

Subject to:

a1, a2, a3, a4 > = 0

T <= T_max

C <= C_max

Q >= Q_min

U >= U_min

x_ij ∈ {0, 1}

Where:

- a1, a2, a3, and a4 are weights representing the relative importance of each objective function.
- T_max, C_max, Q_min, and U_min are predefined thresholds for time, cost, quality, and utilization, respectively.

Quality of Service (QoS) and Resource Allocation

The results also reveal that the QoS is affected by the number of resources in the CMfg system. However, adding up the number of resources to improve the QoS will increase the costs. To counterbalance these positions, the heavy traffic limit approach is suggested. By doing so, it is possible to establish the number of operation machines that are required depending on the QoS.

Methodology

Research design and approach

This study uses quantitative research and is based on the formulation and assessment of mathematical models used in the determination of network traffic in cloud computing networks. Indeed, due to the nature of the problem, which forms the basis of the research, real numeracy data will be involved, which would entail the constitution of models that are predictive in nature; therefore, the perfect methodology for use will have to be quantitative. This research utilizes time series analysis, machine learning and stochastic models to construct hybrid models.

The research utilized data collection, model development, model validation, and model evaluation as guidelines for the research design. At every stage, it is important to determine whether the developed models are reliable and valid in diverse cloud computing scenarios. The study will collect datasets of cloud traffic that are readily available in the public domain; thus, all the models will be trained and tested on real datasets. These data are used in the calibration and testing of mathematical models that are set out based on which evaluation stand Y is judged on a set of parameters, including the Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and R-squared (R^2).

Data collection and preprocessing

This means that the data collection process is a fundamental movement for creating effective predictive models. This research employs datasets of cloud traffic that are in the public domain and can be obtained from AWS, Microsoft Azure, or GCP. Such datasets often contain line-by-line records of all activities occurring in the network though factors such as traffic intensity, packet size, delay, and bandwidth. The collected data will then have to pass through a number of preprocessing steps to be fit for model construction. Preprocessing will include:

Data cleaning: It is important to erasure any data that have been missing, duplicated or incorrectly recorded to influence the outcomes. This may include some variables that have missing values as well as records with missing values, and this may require the imputation of missing data or elimination of the records.

Normalization: Standardizing the data is important so that they can write all variables on the same scale and hence contribute equally to the model. This approach is critical when using variables that are measured on different scales.

Feature selection: Feature selection for predicting traffic occurrence in the network with qualification of the best features out of many features. This may also include correlation analysis or can work with techniques such as Principal Component Analysis (PCA) to work with dimensions. The given dataset was split into a training set and a testing set at a normal ratio of 4:1. The training set is used in model development, while the test set is used in model assessment. The preprocessing step is critical for obtaining better results and helping models learn from the data and make accurate predictions.

Model evaluation and validation

Performance evaluation and validation of the models are essential to ensure that the models that were developed will be capable of correctly estimating network traffic in real-life cloud computing systems. The models are evaluated based on several performance metrics:

Mean Absolute Error (MAE): The MAE is the average of the sums of the absolute differences between the predicted and actual traffic values. It offers a simple point of measure of how accurate a prediction is.

Root Mean Square Error (RMSE): The RMSE is a more sensitive technique for calculating mean squared errors because it squares its errors before averaging them, hence providing large errors and a large proportion of total mean errors. This metric is helpful, especially when large prediction errors are even counterproductive for the model's performance.

R-squared (R^2): This variable defines the extent to which the variance in the dependent variable can be explained by the variance in the independent variables. A higher R^2 value simply means that the proposed model has more capability of accounting for variation in network traffic.

Prediction intervals: For stochastic models, prediction intervals are computed to obtain a prediction interval within which the actual traffic is expected to reach a given confidence level. This shows some extent of uncertainty in the model.

These models will be tested using a test subset of the dataset that was not used during the training phase of the models. This approach helps avoid cases in which the models tend to be very much trained on the training dataset and therefore are not in a position to perform well on other unseen data.

Furthermore, the use of the k-fold cross-validation approach will be applied to further validate the models. This occurs when the data

are divided into k parts, where k is the number of partitions. The model is then trained on k-1 partitions and tested on the other remaining partitions. The above process is then repeated k times, where each of the split sets is used once for the test set.

Ethical considerations and limitations

This research will be performed in an ethical manner to avoid misuse of the information and the modeling procedures. The security of the data will be an issue of most focus, especially because the experiments will involve the use of real cloud traffic datasets, which may involve people's data. Any user and organizational identification will be removed from all the data because of user and organizational confidentiality. Additionally, the research will adhere to data protection laws in the case of the area of study being in the European Union, where data protection is governed by the GDPR. The limitations of the study should also be recognized. A strength may be considered as a weakness, which is that all the datasets were collected from public data sources and hence may be limited in terms of the range of cloud environments. Furthermore, some models require high computational power and thus are likely to be less efficient in terms of real-time training and deployment. These are some of the limitations that will be elaborated upon in the last part of the paper together with research recommendations.

Results and Discussion

This section will reveal the results obtained from the research studies and discuss the consequences of these results for predicting network traffic in the context of cloud computing. Model performance, comparisons of different models and what means to cloud traffic management are usually presented in the Results and Discussion section.

Model performance results

The performances of the developed models, which include time series, machine learning, and hybrid models, were tested using the test dataset. The performance is dissimilar and possesses different levels of accuracy and predictive capability, depending on the evaluation metrics used.

ARIMA model performance: Although the ARIMA model was able to capture long-term trends and periodicity of network traffic, it was less effective at dealing with nonlinear cloud traffic characteristics. Thus, the model we proposed obtained an MAE of 0.015, and an RMSE of 0. The F value is calculated as 025, and the R^2 value is 0. 75. These results indicate that ARIMA has good performance in terms of the trends that can be represented by linear models; moreover, this algorithm has a limited ability to address nonlinear interactions, which are typical of network traffic.

The LSTM model performance: LSTM network performed better than the other network architectures at capturing the short-term and long-term information in the dataset. This means that an MAE of 0 can be achieved when recognizing different elements of the statistical image. 010, an RMSE of 0.

The hybrid ARIMA-ANN model performance: RMSE of 0.010 for the test set indicates that the weather under these conditions has been predicted with high accuracy and an R^2 value of 0. The LSTM model was more accurate than the ARIMA model, with accuracies of 88% and 67%, respectively, of the total number of test data. This approach made it possible to estimate the increase in traffic more accurately, especially when the traffic followed sequences. As a result, the hybrid of the ARIMA-ANN outperformed both the ARIMA and the ANN, which can be observed through the derived performance metrics. The model was able to obtain an MAE of 0, as shown in the next figure. 60 and a mean absolute error of 34 for 008; the corresponding value from the best model is an RMSE of 0. The coefficient of determination was 0.015, while the R^2, which is the coefficient of variation, was equal to 0. 90. This combined method was thus successful in capturing the linear and nonlinear trends in the network traffic data, hence improving the prediction results.

Stochastic model performance: This task showed that stochastic models, mainly those based on queuing theory, were useful because they offered insights into the probabilistic characteristics of network traffic. Although the prediction intervals were registered as wider, thus implying greater variability in the outcomes, the mean of the prediction was still predetermined to be within the range of acceptable error. For this model, the MAE reached 0, and the RMSE was also equal to 0. 012 and 0. In addition, there is a moderated dualistic relationship between L7 and L9, which are equal to 020, and the R^2 value is less than 0.82.

These findings help provide insights into the performance of hybrid and machine learning-based algorithms in forecasting network traffic in cloud computing systems. When comparing the performance of the time series and the machine learning methods, the latter achieved the best result, proving the effectiveness of using both techniques in modeling cloud traffic.

Heavy traffic limit theory and QoS classes

Based on the waiting time, four distinct QoS classes are considered:

1. **Zero-Waiting-Time (ZWT):** Tasks are executed immediately upon arrival.
2. **Minimal-Waiting-Time (MWT):** Tasks experience minimal waiting time.
3. **Bounded waiting time (BWT):** Tasks have a bounded waiting time.
4. **Probabilistic-Waiting-Time (PWT):** Tasks have a probabilistic waiting time.

To analyze these QoS classes under heavy traffic conditions, the following mathematical formulations are used:

Mathematical Formulations

1. Zero waiting time (ZWT):

$$\lim(n\to\infty)\ P(N \geq n) = 0$$

Where:

- N is the total number of tasks
- n is the number of operational machines

2. Minimal Waiting Time (MWT):

$$\lim(n\to\infty)\ P(N \geq n) = \alpha$$

Where:

- α is a constant $(0 < \alpha < 1)$

3. Bounded Waiting Time (BWT):

$\lim(n\to\infty) P(N \geq n) = 1$

$\lim(n\to\infty) P(W \geq t1) = \sigma\_n$

$\lim(n\to\infty) \sigma\_n = 0$

Where:

- W is the waiting time
- t1 is the waiting time threshold
- σ_n is the rate of decrease

4. Probabilistic Waiting Time (PWT):

$\lim(n\to\infty) P(N \geq n) = 1$

$\lim(n\to\infty) P(W \geq t2) = \sigma$

Where:

- t2 is the waiting time threshold
- σ is a constant ($0 < \sigma < 1$)

Heavy Traffic Limit Analysis

Under heavy traffic conditions (traffic intensity approaches 1), the following relationships hold:

1. ZWT:

$\lim(n\to\infty) (1 - \rho\_n)^n = 0$

Where:

- ρ_n is the traffic intensity

2. MWT:

$\lim(n\to\infty) (1 - \rho\_n)^n = \beta$

Where:

- β is a constant

3. BWT:

$\lim(n\to\infty) (1 - \rho\_n)^{(-\ln(\sigma\_n))} = \tau$

$\lim(n\to\infty) \sigma\_n * \exp(kn) = \infty$

Where:

- τ is a constant
- k is a constant

4. PWT:

$P(W \geq t2) \approx \exp(-2n\mu(1-p)\, t2/(1+c^2))$

$\lim(n\to\infty) (1 - \rho\_n)^n = \gamma$

Where:

- γ is a constant

These equations provide guidelines for determining the required number of machines to meet specific QoS requirements in a cloud environment.

Interpretation of the results

This work is substantial in its ability to guide the further management of cloud traffic. The enhanced accuracy of the performance levels obtained by LSTM and the hybrid model of ARIMA and ANN strongly suggest that these approaches will be highly appropriate for modeling and predicting traffic within more volatile and unpredictable networks. The forecasts that they provide for traffic patterns can prove beneficial for cloud service providers because they can plan and allocate their resources better and thus reduce the latency and enhance the overall quality of service.

Implications for cloud resource management: Traffic forecasting helps cloud providers allocate resources for traffic needs so as not to lack capacity at some point or underprovide. This can lead to IR savings, efficient utilization of resources and especially during hours of high uptake during which quality services are offered. Because the LSTM and hybrid models correctly predict real-time traffic flows, the software is useful for real-time traffic observation and control.

Scalability and adaptability: The flexibility of machine learning algorithms such as Long Short-Term Memory (LSTM) is crucial in cloud settings due to constant changes in traffic characteristics, such as variations in users' traffic intensity or the development of new services. These models can be regularly retrained for the purpose of adjusting to the current conditions, hence making the projections highly relevant.

Potential challenges: As effective as these models may be, their structural and computational concerns may present difficulties in real-time scenarios. As demonstrated, hybrid models generally outperform pure exploitation and exploration models, which makes them more accurate; however, this accuracy must be attained with caution concerning overfitting and validation. Furthermore, stochastic models that are less accurate reflect the importance of accounting for uncertainty in traffic estimates, especially where the physical environment is unpredictable.

Future research directions: Based on these results, the following research questions can be proposed for future research: One of the further development paradigms is the use of another more automated hybrid structure that involves additional artificial intelligence methods, for example, reinforcement learning, to minimize the error rate. One of the further research directions is the testing of these models under different types of cloud settings, whether edge computing or multiple clouds, to evaluate their performance under certain conditions.

Broader impact: The success of these predictive models in a cloud environment could extend to other domains that require prediction of network traffic, such as telecommunications, cybersecurity and smart cities. These techniques can be applied to such contexts as the procedures herein could be useful in the purposeful management of resources in a multitude of applications relating to telecommunication networks.

Conclusion

This research aimed to design and assess mathematical models for forecasting network traffic in a cloud computing environment. The study was centered on the analysis of the efficiency of standard time series models and machine learning algorithms, as well as the integration of these methods, to obtain a clear understanding of the

effectiveness of models in traffic forecasting. The results showed that even though linear time series models such as ARIMA provide good fits for linear traces and seasonality, they lack the flexibility required to model the nonlinearity of cloud traffic. Another key difference was that traditional models, especially the machine learning models, which include long short-term memory (LSTM) networks, were found to be comparatively more effective at capturing both short- and long-term dependencies, which helped in obtaining more accurate estimations. The best model that was formulated was the hybrid model of the ARIMA-ANN model, in which the features of both techniques were used, and the results proved most effective in forecasting network traffic. The above findings are very useful for cloud service providers since predictive models can be used to maximize the resources needed to minimize latency and enhance the quality of the services being offered. This is because accurate traffic prediction and modeling solutions can assist providers in effectively monitoring their infrastructure and administering resources at high service availability levels. Moreover, the research also showed how uncertainty must be integrated into traffic prediction problems through the principles proven in stochastic models. The potential for traffic outcomes is often vital for planning, especially in the case of networks that exhibit unstipulated or highly fluctuating behavior.

Conflict of interest

Author's declare there is no conflict of interest.

References

1. Aburukba RO, Landolsi T, Omer D (2021) A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices. Journal of Network and Computer Applications 180: 102994. [Crossref] [GoogleScholar]
2. Saxena D, Singh AK (2022) Autoadaptive learning-based workload forecasting in dynamic cloud environment. International Journal of Computers and Applications 44(6): 541-551. [Crossref] [GoogleScholar]
3. Tuli S, Ilager S, Ramamohanarao K, Buyya R (2020) Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. IEEE transactions on mobile computing 21(3): 940-954. [Crossref] [GoogleScholar]
4. Li H, Wang SX, Shang F, Niu K, Song R (2024) Applications of large language models in cloud computing: An empirical study using real-world data. International Journal of Innovative Research in Computer Science & Technology 12(4): 59-69. [GoogleScholar]
5. Lohrasbinasab I, Shahraki A, Taherkordi A, Jurcut DA (2022) From statistical-to machine learning-based network traffic prediction. Transactions on Emerging Telecommunications Technologies 33(4): e4394. [Crossref] [GoogleScholar]
6. Shafiq DA, Jhanjhi NZ, Abdullah A (2022) Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences 34(7): 3910-3933. [Crossref] [GoogleScholar]
7. Abouelyazid M (2022) Forecasting resource usage in cloud environments using temporal convolutional networks. Applied Research in Artificial Intelligence and Cloud Computing 5(1): 179-194.
8. Balarezo JF, Wang S, Chavez KG, Al-Hourani A, Kandeepan S (2022) A survey on DoS/DDoS attacks mathematical modeling for traditional, SDN and virtual networks. Engineering Science and Technology, an International Journal 31: 101065. [Crossref] [GoogleScholar]
9. Omer S, Azizi S, Shojafar M, Tafazolli R (2021) A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers. Journal of systems architecture 115: 101996. [Crossref] [GoogleScholar]
10. Hsieh SY, Liu CS, Buyya R, Zomaya AY (2020) Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. Journal of Parallel and Distributed Computing 139: 99-109. [Crossref] [GoogleScholar]
11. Saeik F, Avgeris M, Spatharakis D, Santi N, Dechouniotis D, et al. (2021) Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. Computer Networks 195: 108177. [Crossref] [GoogleScholar]
12. Tuli S, Tuli S, Tuli R, Gill SS (2020) Predicting the growth and trend of Covid-19 pandemic using machine learning and cloud computing. Internet of things 11: 100222. [Crossref] [GoogleScholar] [Pubmed]
13. Gill SS, Tuli S, Toosi AN, Cuadrado F, Garraghan P, et al. (2020) ThermoSim: Deep learning-based framework for modeling and simulation of thermal-aware resource management for cloud computing environments. Journal of Systems and Software 166: 110596. [Crossref] [GoogleScholar]
14. Alghamdi MI (2022) Optimization of load balancing and task scheduling in cloud computing environments using artificial neural networks-based Binary Particle Swarm Optimization (BPSO). Sustainability 14(19): 11982. [Crossref] [GoogleScholar]