Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Horizon Cloud Resource Demand Prediction using Temporal Fusion Transformer

Praveen Kumar Kollu

praveen@vrsiddhartha.ac.in

Siddhartha Academy of Higher Education (Deemed to be University)

Rajesh Uda

Siddhartha Academy of Higher Education (Deemed to be University)

Raj Manoj Bytari

Siddhartha Academy of Higher Education (Deemed to be University)

Jeevan Kumar Naik Banavathu

Siddhartha Academy of Higher Education (Deemed to be University)

Trinetra Saggurthi

Siddhartha Academy of Higher Education (Deemed to be University)

Research Article

Keywords: Temporal Fusion Transformer, Cloud Resource Allocation, Workload Forecasting, Multi-horizon Forecasting, Attention Mechanism, Auto-scaling

Posted Date: May 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9748050/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Cloud resource management remains a fundamental challenge due to highly dynamic workload patterns and stringent performance requirements. Efficient management of virtual machine resources is essential to ensure optimal performance while minimizing operational costs. Traditional reactive auto-scaling approaches struggle to handle abrupt workload variations, leading to resource inefficiency and performance degradation. This study proposes a proactive resource management approach using multi-horizon time-series forecasting. The Alibaba Cluster Trace 2017 dataset, which contains large-scale real-world cloud workload data including CPU, memory, and disk utilization, is utilized for analysis. A Temporal Fusion Transformer (TFT) model is employed to predict future CPU utilization at multiple time horizons (1, 5, and 15 minutes). The TFT model integrates recurrent neural networks, attention mechanisms, gating layers, and variable selection networks to effectively capture complex temporal dependencies while providing interpretability. The proposed approach enables the identification of key factors influencing resource utilization and supports effective decision-making for auto-scaling strategies. Experimental results demonstrate that the TFT model achieves strong forecasting performance, with high accuracy in short-term predictions and stable behavior across longer horizons. These findings highlight the effectiveness of interpretable deep learning models in enabling proactive cloud resource management and improving the reliability of cloud infrastructures.

1 Introduction

Cloud computing refers to a new computing model, which provides computing internet-based services like servers, storage, databases, networking, and software [26, 28]. It allows organizations to remotely store data with the benefit of being scalable and cost efficient through dynamic adjustment of resources according to the demand [27]. One of the most important features of cloud computing is the use of Virtual Machines (VMs), whereby several operating systems can be used in the same physical host using the method of virtualizing [30]. Every VM has a virtual CPU and memory and storage resources and the management of the resources and projection of the performance is essential to achieve maximum efficiency [29].

Auto-scaling is a necessary aspect in the cloud environment that enables the system to actively balance the number of computational resources in real time according to the workload requirements [2, 26]. Nevertheless, conventional auto-scaling solutions are frequently reactive and they are based on pre-determined thresholds, and this restricts their usefulness in dynamic contexts [2]. The nature of cloud resource usage is temporal and it is reflected as time series data, which is a collection of observations recorded periodically e.g. CPU, memory, disk, network usage [25]. The time series forecasting is important in estimating future resource requirement and proactive allocation of resources [3, 29].

Time series prediction has been popularly performed using deep learning models [24]. Long Short-term Memory (LSTM) networks are especially useful in sequential data and have been trained in large scale to predict the cloud workload [9, 10, 12]. Hybrid methods like CNN-LSTM are also an improvement on

predictive accuracy since they extract both the spatial and temporal features [8, 13]. In more recent times, another strong architecture that has been developed is the Temporal Fusion Transformer (TFT) as a system of multi-horizon forecasting [1, 4]. It is a combination of recurrent networks and attention mechanisms that model complex time-related relationships and can be interpreted [1, 19–21].

The workload patterns have turned out to be increasingly dynamic and unpredictable with the fast growth of the cloud-based applications. The older reactive auto-scaling systems tend to react slowly to the abrupt workload variations resulting in either resource over-provisioning or performance issues [2]. This has motivated the necessity of anticipatory and proactive resource management techniques.

Despite the fact that LSTM-based models have been shown to be more accurate in forecasting, they are not as effective in complex multi-variable dependence and long-range temporal relationship. The recent research has tried to increase the prediction rates by making use of ensemble learning, anomaly processing and hybrid constructions [6, 7]. Also, uncertainty-sensitive forecasting methods have been presented in order to enhance reliability in dynamic clouds [3].

The state-of-the-art cloud systems need multi-horizon forecasting to forecast the demand for resources at different future time steps concurrently. Transformer models like TFT have also been shown to outperform other models in terms of capturing both short and long-term dependencies as well as be interpretable, which makes them highly suitable in these tasks [1, 4]. These innovations encourage the creation of a strong and interpretable forecasting model to predict the demand of cloud resources.

Although cloud resource management has made tremendous progress, it is quite difficult to predict the demand of resources because cloud workloads are highly dynamic and non-linear. Standard reactive auto-scaling methods cannot meet the needs of fast-changing workloads, and traditional machine learning models do not always capture complex temporal relationships [2, 10].

Current deep learning methods including LSTM and its derivatives improve prediction accuracy but still lack multi-horizon prediction capability and interpretability [9, 10]. Additionally, most models struggle to handle multivariate time series where multiple resource metrics interact simultaneously [6, 18]. Advanced methods using attention and ensemble learning achieve better performance but often increase computational complexity [7, 18].

Therefore, there is a need for an accurate, scalable, and interpretable forecasting model capable of performing multi-horizon predictions in cloud environments and enabling proactive resource planning without requiring infrastructure changes.

The primary objectives of this work include preprocessing large-scale cloud workload data, performing feature engineering by categorizing static and dynamic features, developing a Temporal Fusion Transformer (TFT) model for capturing temporal dependencies, performing multi-horizon forecasting of CPU utilization, and improving prediction accuracy and interpretability using attention-based deep learning techniques [1, 4, 18].

This study focuses on predicting cloud resource usage using historical time series data without implementing real-time auto-scaling mechanisms. The model leverages large-scale datasets containing CPU, memory, disk, and workload features, and applies deep learning techniques such as LSTM and TFT for forecasting [1, 10]. The system supports multivariate and multi-horizon prediction, which is essential for modern cloud environments [6, 18]. However, the current scope is limited to offline analysis and does not include real-time deployment or integration with cloud infrastructure, which can be explored in future work.

2 Literature Review

Cloud resource management and workload forecasting have received extensive research as the demand for efficient and scalable cloud infrastructures grows, and the cost of cloud computing is of particular concern. Transformer models have been emerging in the limelight and Lim et al. [1] have proposed the Temporal Fusion Transformer (TFT) that combines LSTM, attention, and variable selection networks to make precise and explainable multi-horizon predictions. More advancements were also illustrated by Zhang et al. [4], who integrated TFT with multi-task learning to improve the prediction performance especially in a situation where less data are available. Conversely, Buyya et al. [2] thoroughly discussed the traditional cloud auto-scaling strategies and identified scaling methods as reactive and proactive and the shortcomings of threshold-based methods in dealing with dynamic workloads.

To enhance the reliability of the predictions, Iqbal et al. [3] suggested uncertainty-sensitive forecasting based on Bayesian deep and transfer learning, which allows making strong predictions with different cloud conditions. On the same note, Kertesz et al. [5] proposed a machine learning-based autoscaling system, which includes statistical feature selection and proactive scaling, which shows enhanced Quality of Service (QoS) in actual cloud systems. Hasan et al. [6] also made improvements on the forecasting systems where they suggested a multi-time series forecasting framework, which integrates clustering, anomaly detection, and machine learning models to achieve optimal resource exploitation and cost-effectiveness.

The recent studies have also concerned the enhancement of the long-term forecasting accuracy and strength. Kim et al. [7] proposed a structure of ensemble that combines transformer models and linear prediction methods with anomaly management to increase the stability of the predictions in multivariate setting. Hybrid deep learning systems have proven to be a promising option; Patil and Sahoo [8] and Ouhame et al. [13] suggested CNN-LSTM models, which have proven to be good at ensuring that both spatial and temporal features of the workload data are captured and, therefore, prediction accuracy and SLA violations are reduced. Moreover, a LSTM-based Stackelberg game-theoretic model proposed by Hossain et al. [9] that maximizes the cloud resource allocation by simulating interactions between service providers and users showed a better fairness and efficiency of resource allocation.

The proactive resource allocation has also been extensively investigated using LSTM-based predictive modeling. Verma and Kaushal [10] designed an LSTM-based scheme using scheduling schemes to

optimize resource allocation, and Thonglek et al. [12] confirmed the usefulness of LSTM models in optimizing the use of resources in a data center. Jeong et al. [11] also suggested additional enhancements by providing a resource management system based on deep neural networks to allow stable and efficient cloud operations. Ensemble and hybrid methods, like random forest and LSTM combination by Valarmathi and Raja [15] and attention-based hybrid LSTM by Bi et al. [18], have additionally enhanced the level of prediction and strength against cloud settings.

Furthermore, the recent publications like Zi [14] focused on the application of recurrent neural networks in forecasting the workload of the clouds in time series, and Yuan et al. [16] suggested a better LSTM-based model that can be applied to large-scale data centers with enhanced generalization. In the cloud machine learning systems, Paripati [17] emphasized the need to optimize the scaling and resource exploitation of the system. Regardless of such improvements, the current methods usually aim at either accuracy of prediction or a resource allocation strategy without having a common framework that is suitable to offer not only interpretability but also effective multi-horizon forecasting. Recent transformer-based forecasting models such as long-term sequence transformers and neural basis expansion approaches have also shown promising results in time series prediction [22, 23].

Based on the above studies, our project uses the Temporal Fusion Transformer (TFT) to make correct and understandable predictions of multi-horizon cloud resource demand. In contrast to the deployment-oriented solutions, our solution is based on predictive modeling without explicitly changing cloud infrastructure, which offers a base of intelligent and proactive planning of the resources.

3 Methodology

The proposed system forecasts the demand of cloud resources in a multi-horizon with the help of a Temporal Fusion Transformer (TFT) model. The methodology takes a systematic path consisting of data preparation, preprocessing, model design, and implementation pathway. The method is based on the recent progress in time series predictions using deep learning and managing cloud resources [1, 6, 18].

3.1 Architectural Design

The proposed system architecture will be configured to process multi-horizon time series data of large scale and produce correct multi-horizon predictions. The system has various components that are connected together as shown in the system architecture diagram.

The data ingestion layer receives data on cloud workload of various sources including server usage logs, machine events, and batch workload traces. The preprocessing layer includes the time stamps alignment, missing value, normalization and feature engineering. The idea behind the use of feature engineering is that inputs are split into both classes of features, namely, static and dynamic features, which are common to complex forecasting models such as TFT [1].

The obtained processed data is converted to sequences through a sliding window mechanism and sent through the TFT model layer that combines LSTM-based temporal encoding with multi-head attention mechanisms. The combination allows the model to be sensitive to both short-term and long-term dependencies, as hybrid deep learning methods applied in cloud forecasting do [8, 13]. Lastly, the prediction layer creates multi-horizon predictions whereas the evaluation module compares the performance measures, which include MAE, MSE, and R^2.

This architecture provides scalable and readable forecasting, which is consistent with the current cloud resource prediction system [6, 18].

3.2 Dataset Preparation

The dataset applied in this work is Alibaba Cluster Trace 2017 dataset that includes real-life information about cloud workloads that are collected in the production settings. Cloud workload forecasting research is typically based on such large-scale datasets to make models realistic [7].

The data contains server related usage statistics (CPU, memory, disk), machine event log and batch workload statistics. These non-homogenous data sets are synchronized with timestamps to create a combined multivariate time series data.

Key preprocessing steps include the removal of redundant and irrelevant attributes, aggregation of workload features from batch tasks, resampling into fixed 1-minute intervals, handling missing values using forward and backward filling, and normalization of features to improve model convergence.

These steps are consistent with preprocessing techniques adopted in prior studies for improving forecasting accuracy in cloud environments [6, 7].

To ensure uniform feature scaling and improve model convergence, Min-Max normalization is applied:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \#(1)$$

3.3 Model Architecture

The given model is founded on Temporal Fusion Transformer (TFT) that is specifically created on interpretable multi-horizon forecasting [1]. The TFT architecture is an architecture that unites recurrent neural networks with attention to successfully establish temporal dependencies.

The data of the cloud resource is in the form of multi-horizon time series:

$$X = \{x_1, x_2, x_3, \ldots, x_t\} \#(2)$$

where every x_t is the resource characteristics like CPU, memory and disk utilization.

The model consists of an input layer that accepts multivariate time series sequences, followed by a Bidirectional LSTM encoder that captures temporal dependencies in both forward and backward directions, similar to LSTM-based forecasting approaches [10, 12]. The LSTM cell updates are defined as:

$$f_t = \sigma\left(W_f\left[h_{\{t-1\}}, x_t\right] + b_f\right) \#(3)$$

$$i_t = \sigma\left(W_i\left[h_{\{t-1\}}, x_t\right] + b_i\right) \#(4)$$

$$\widetilde{C_t} = \tanh\left(W_C\left[h_{\{t-1\}}, x_t\right] + b_C\right) \#(5)$$

$$C_t = f_t \odot C_{\{t-1\}} + i_t \odot \widetilde{C_t} \#(6)$$

$$o_t = \sigma\left(W_o\left[h_{\{t-1\}}, x_t\right] + b_o\right) \#(7)$$

$$h_t = o_t \odot \backslash tanh\left(C_t\right) \#(8)$$

Layer normalization is applied to stabilize learning. The model further incorporates a multi-head attention mechanism that focuses on relevant time steps, improving interpretability and performance [1, 8]. The attention mechanism in the model is defined as:

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \#(9)$$

Multi-head attention is computed as:

$$MultiHead\left(Q, K, V\right) = Concat\left(head_1, \ldots, head_h\right)W^O \#(10)$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \#(11)$$

A feature fusion layer combines temporal and contextual representations, followed by a fully connected decoder that applies nonlinear transformations. Finally, the output layer predicts CPU utilization for multiple horizons.

The model performs multi-horizon forecasting using:

$$\widehat{Y}_{\{t+h\}} = f\left(X_{\{t-k\}}, X_{\{t-k+1\}}, \ldots, X_t\right),\ h \in \{1,5,15\} \#(12)$$

where $h \in \{1,5,15\}$ represents different prediction horizons.

This is vital to the active resource allocation and is one of the main merits of transformer-based models compared to the traditional LSTM methods [4, 18].

As shown in Fig. 1, the overall model structure illustrates the integration of LSTM-based temporal encoding with multi-head attention mechanisms for multi-horizon cloud resource demand prediction.

3.4 Training Procedure

Temporal Fusion Transformer (TFT) model is trained on historical data on cloud workloads to learn temporal pattern and forecast precise multi-horizon workloads. The training is done in a systematic manner in order to achieve stability, generalization and effective convergence.

The multivariate time series data that has been preprocessed is first converted to input-output sequences with a sliding window mechanism. All inputs are historical measurements during a determined look-back (e.g., 60 time steps) whereas the output is the future measurements of CPU utilization at different horizons (+ 1, + 5 and + 15 minutes).

Gradient based learning is used to optimize the model parameters. Training is aimed at reducing the difference between real and approximated values. The training loss used is the Huber loss that is expressed as follows:

$$L_{\delta}\ \left(y,\hat{\mathrm{y}}\right)=\begin{cases}\frac{1}{2}\left(y-\hat{\mathrm{y}}\right)^{2} & if\ \left|y-\hat{\mathrm{y}}\right|\leq\ \delta\\ \delta\ \left|y-\hat{\mathrm{y}}\right|-\frac{1}{2}\delta^{\,2} & otherwise\end{cases}\#\left(13\right)$$

The Huber loss is preferred as it is insensitive to outliers and sensitive to small errors hence it is suitable to real-world data of cloud workloads.

Adam optimizer is used to update the model parameters and the learning rate is adjusted in the process of training to converge faster. The rule of updating the parameter is as follows:

$$m_t=\beta_1 m_{\{t-1\}}+\left(1-\beta_1\right)g_t\ \#\left(14\right)$$

$$v_t=\beta_2 v_{\{t-1\}}+\left(1-\beta_2\right)g_t^2\ \#\left(15\right)$$

$$\widehat{m_t}=\frac{m_t}{1-\beta_1^{\,t}}\ \#\left(16\right)$$

$$\hat{v}_t=\frac{v_t}{1-\beta_2^{\,t}}\ \#\left(17\right)$$

$$\theta_t=\theta_{t-1}-\frac{\alpha\,\widehat{m_t}}{\sqrt{\hat{v}_t}+\epsilon}\ \#\left(18\right)$$

with gradient of the loss function g_t, learning rate α, and decay rates β_1, β_2.

In order to avoid overfitting and enhance generalization, dropout and L2 regularization are used in the fully connected layers. There is also early stopping which is done on the basis of validation loss to end training when there is no more improvement.

Learning rate scheduling is also included in order to dynamically change the learning rate in training so that convergence is not unstable and local minima are not reached.

Mini-batch gradient descent is used to train the model in numerous epochs. The training data is fed through the model to obtain the predictions and calculate the loss and backpropagate it to the model weights during every epoch.

Once the model has completed its training, it is tested on a different test dataset based on performance metrics to determine the level of prediction performance using Mean Absolute Error, Mean Squared Error, and R-squared score.

Such a training process allows the model to be able to learn intricate temporal relationships and make credible multi-horizon predictions on cloud resource demand.

3.5 Implementation Flow

The process of implementing the suggested system is based on a systematic pipeline which converts raw cloud workload data into a multi-horizon prediction. The workflow will have the correct data processing, feature learning and training of a model to make correct predictions.

First, the data on cloud workloads is collected using the Alibaba Cluster Trace 2017 dataset where the measurements of resource usage are recorded as time-stamps of CPU, memory, and disk usage. The raw data is then pre-processed which includes handling missing values, timestamp alignment and normalization of features to ensure consistency.

After the preprocessing, feature engineering is undertaken in order to identify meaningful static and dynamic features. Machine capacity is a static feature and the use of CPU and workload characteristics are dynamic features because they vary over time.

The processed data is transformed into input-output sequences based on a sliding window mechanism in order to allow time series forecasting. The model is able to learn the temporal relationships using past observations using this approach.

The input sequences are characterized as:

$$X_t = [x_{t-k}, x_{t-k+1}, \dots, x_t] \quad \#(19)$$

where *k* represents the look-back window size (e.g., 60 time steps).

The corresponding output targets for multi-horizon forecasting are defined as:

$$Y_t = \left[y_{t+1}, y_{t+5}, y_{t+15}\right] \quad \#(20)$$

where the model predicts CPU utilization at future time steps of + 1, +5, and + 15 minutes.

The overall forecasting task can be expressed as:

$$\widehat{Y}_{\{t+h\}} = f\left(X_{\{t-k\}}, X_{\{t-k+1\}}, \ldots, X_t\right),\ h \in \{1,5,15\}\ \#(21)$$

where *f*(·) represents the Temporal Fusion Transformer model and *h* ∈ {1,5,15} denotes the prediction horizons.

Once the sequence has been generated, it is divided into training, validation and testing sets in chronological order in order to maintain temporal dependencies. The model is subsequently trained with the use of historical sequences and assessed with the help of conventional performance measures.

Lastly, the trained model can be used to formulate multi-horizon predictions that can be utilized in proactive cloud resource management and decision-making.

3.6 Evaluation Metrics

To achieve accurate and reliable predictions, the proposed cloud workload forecasting system is tested with the help of a combination of common regression metrics to assess its performance. The predicted output is compared to the actual values of CPU utilization in the dataset to determine the level of accuracy and effectiveness of the model at various forecasting horizons.

The most outstanding evaluation measure is the Mean Squared Error (MSE), which is described as:

$$MSE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(y_i - \hat{\mathrm{y}}_i\right)^2\ \#(22)$$

which calculates the mean squared error between measured and estimated values but with more weight on larger errors. Root Mean Squared Error (RMSE) is determined to represent the error of the target variable in the same units:

$$RMSE = \sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(y_i - \hat{\mathrm{y}}_i\right)^2}\ \#(23)$$

which provides a more interpretable measure of prediction error. The Mean Absolute Error (MAE) is also used, defined as:

$$MAE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\left|y_i - \hat{\mathrm{y}}_i\right|\ \#(24)$$

which measures the average magnitude of prediction errors regardless of direction and is less sensitive to

outliers compared to MSE.

To evaluate the goodness of fit of the model, the coefficient of determination (R^2) is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \# (25)$$

which is used to determine the percentage variance of the actual values that is accounted for by the model predictions.

The model is also further analyzed in terms of performance at various forecasting horizons, which are + 1 minute, + 5 minutes, and + 15 minutes. It can be observed that the model is more accurate for short-term predictions, while performance gradually declines as the horizon becomes longer due to the increasing uncertainty in future workload patterns.

All these measures give a full assessment of the model in terms of prediction accuracy, magnitude of errors, and explanatory power, and prove the effectiveness of the suggested system for proactive cloud resource management and multi-horizon prediction.

4 Results and Analysis

This section presents the performance analysis of the proposed multi-horizon cloud resource demand prediction model based on the Temporal Fusion Transformer (TFT). The Alibaba Cluster Trace 2017 dataset is used to obtain the results and evaluate them from multiple perspectives, including prediction accuracy, error distribution, model stability, and forecasting reliability.

4.1 Quantitative Performance Evaluation

Standard regression metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 score are used to quantitatively evaluate the performance of the proposed Temporal Fusion Transformer (TFT) model across different forecasting horizons. In addition, percentage-based and scaled error metrics including Mean Absolute Percentage Error (MAPE), symmetric MAPE (sMAPE), and Mean Absolute Scaled Error (MASE), along with prediction bias, are considered to provide a comprehensive evaluation.

Table 1
Extended multi-horizon forecasting performance metrics

Horizon	MAE	RMSE	*R*2	MAPE	sMAPE	MASE	Bias
+ 1	0.2731	0.3281	0.6104	19.77	20.70	4.1318	-0.2632
+ 5	0.4661	0.5319	-0.1031	39.67	42.26	8.6067	-0.4598
+ 15	0.4253	0.4916	0.0269	40.65	34.64	8.9658	-0.4074

Table 1 shows that the best performance is achieved at the short-term forecasting horizon of + 1 minute, with the lowest error values (MAE = 0.2731, RMSE = 0.3281) and the highest R^2 score (0.6104). The

relatively low MAPE (19.77%) and sMAPE (20.70%) further confirm the accuracy and reliability of short-term predictions.

As the forecasting horizon increases to + 5 minutes, the prediction error increases significantly, with MAE rising to 0.4661 and RMSE to 0.5319. The negative R^2 value (-0.1031) indicates reduced predictive capability at this horizon. Similarly, higher MAPE (39.67%) and sMAPE (42.26%) values reflect increased uncertainty in medium-term predictions.

For the + 15 minute horizon, the model shows slight improvement over the + 5 horizon in terms of R^2 (0.0269), although the error metrics remain relatively high. This suggests that while the model captures general workload trends, prediction precision decreases for longer horizons.

The slight improvement at the + 15 minute horizon compared to the + 5 minute horizon may be attributed to the smoothing of workload fluctuations over longer time intervals, which reduces short-term variability and allows the model to capture broader temporal patterns more effectively.

The negative bias values across all horizons indicate a tendency of the model to slightly underestimate CPU utilization. Additionally, higher MASE values at longer horizons highlight reduced forecasting efficiency compared to naive baseline methods.

Overall, these results demonstrate that the proposed TFT model is highly effective for short-term forecasting while maintaining moderate predictive capability for longer horizons. The model demonstrates strong performance for short-term forecasting, while moderate to limited accuracy is observed for longer horizons, which is consistent with the inherent complexity and uncertainty of cloud workload dynamics.

4.2 Comparison with Existing Methods

To further evaluate the effectiveness of the proposed Temporal Fusion Transformer (TFT) model, its performance is compared with commonly used deep learning models for time series forecasting, including Long Short-Term Memory (LSTM) and hybrid CNN-LSTM models, which are widely adopted in cloud workload prediction

[8, 10, 13].

Traditional LSTM models are effective in capturing sequential dependencies but often struggle with longterm temporal relationships and multivariate feature interactions. Similarly, CNN-LSTM models improve feature extraction by combining spatial and temporal representations; however, they still have limitations in handling multi-horizon forecasting and interpretability.

In contrast, the proposed TFT model leverages attention mechanisms and variable selection networks to capture both short-term and long-term dependencies while providing interpretable predictions. The experimental results demonstrate that the TFT model achieves superior performance in short-term

forecasting (+ 1 minute horizon), with lower error metrics and higher R^2 values compared to conventional approaches reported in the literature.

Although the performance degrades for longer horizons (+ 5 and + 15 minutes), this trend is consistent with observations in existing studies, where prediction uncertainty increases with forecasting horizon. However, the TFT model maintains competitive performance while offering the additional advantage of interpretability and multi-horizon prediction capability, which are not effectively addressed by traditional LSTM-based models.

Overall, the comparison indicates that the proposed TFT model provides a more robust and scalable solution for cloud workload forecasting, particularly in scenarios requiring proactive resource allocation and multi-step prediction. This demonstrates the practical advantage of TFT in real-world cloud environments where both accuracy and interpretability are critical.

4.3 Prediction vs Actual Plot

In this plot, the X-axis represents actual values from the dataset, while the Y-axis represents predicted values from the model. A straight diagonal line indicates a perfect prediction.

As shown in Fig. 2, the predicted values closely follow the actual CPU utilization, indicating strong model accuracy.

4.4 Error Distribution Analysis

In this analysis, the X-axis represents the prediction error, calculated as the difference between the actual CPU utilization and the predicted CPU utilization, while the Y-axis represents the frequency of occurrence of prediction errors within a specific range. A symmetric distribution centered near zero implies unbiased predictions, while wider distributions at higher horizons highlight the natural trade-off between forecast horizon and prediction accuracy.

As illustrated in Fig. 3, the error distribution is centered near zero, demonstrating unbiased prediction performance.

4.5 Temporal Stability of Forecast Errors

In this analysis, the X-axis represents the sequence index, indicating the progression of time-series prediction windows across the test dataset, while the Y-axis represents the rolling Mean Absolute Error (MAE), computed over a fixed window to assess temporal stability of prediction errors. The relatively smooth and bounded error trends across all horizons indicate that the model maintains temporal stability, while the gradual increase in error magnitude highlights the expected degradation in accuracy for longer-term forecasts.

As shown in Fig. 4, the model maintains stable error trends across different forecasting horizons.

4.6 Prediction with Confidence Band (+ 1 Minute)

In this plot, the X-axis represents the sequence index corresponding to successive time steps in the test dataset, while the Y-axis represents CPU utilization values, including actual measurements and model predictions. Occasional deviations outside the confidence interval correspond to abrupt workload changes, suggesting areas where adaptive or probabilistic modeling could further enhance robustness.

As depicted in Fig. 5, most predicted values lie within the confidence interval, indicating reliable forecasting performance.

4.7 Performance on Alibaba Cluster Trace 2017 Dataset

This presents the comparison analysis between the real and projected values of CPU utilization realized through the application of the proposed Temporal Fusion Transformer model on the Alibaba Cluster Trace 2017 dataset. The predicted CPU utilization values by the TFT model, indicated by the blue dotted line, are well correlated with the corresponding actual CPU utilizations, indicated by the blue solid line across different forecasting horizons of + 1, +5, and + 15 minutes. The TFT model effectively captures temporal variations and workload dynamics of cloud infrastructures, with minor deviations observed at longer horizons.

In conclusion, the TFT approach can be effectively used to estimate resource requirements, which can support proactive auto-scaling in cloud infrastructures.

As shown in Fig. 6, the predicted CPU utilization closely aligns with the actual values across multiple forecasting horizons.

5 Conclusion and Future Work

This paper presents an effective deep learning-based solution for optimizing cloud resource management. The proposed approach utilizes the Temporal Fusion Transformer (TFT) architecture to perform multi-horizon forecasting for cloud auto-scaling.

The developed system is capable of accurately predicting future CPU utilization over short-term horizons, enabling proactive scaling of cloud resources. In contrast to traditional threshold-based auto-scaling methods, the proposed approach demonstrates improved performance and prediction accuracy. The architecture integrates robust data preprocessing, feature engineering, and attention-based modeling to capture complex relationships between static virtual machine parameters and dynamic workload patterns.

Furthermore, the model effectively adapts to sudden fluctuations in workload demand, reducing the risk of over-provisioning and minimizing operational costs. The ability to perform multi-horizon forecasting

enhances decision-making by providing insights into future resource requirements, which is essential for highly scalable cloud environments.

Future work may focus on extending this framework to real-time deployment using streaming data for continuous workload monitoring and prediction. Additionally, incorporating probabilistic forecasting techniques into the TFT model can improve robustness under highly volatile workload conditions. Integration of the prediction model with reinforcement learning-based auto-scaling strategies can further enable self-optimizing cloud resource management.

Moreover, the framework can be enhanced for production-level deployment by incorporating automated scaling triggers and interactive data visualization dashboards, enabling efficient and intelligent cloud infrastructure management.

Declarations

- **Funding:** The authors did not receive any specific funding for this research.
- **Conflict of Interest:** The authors declare that they have no conflict of interest.
- **Ethical Approval:** This article does not contain any studies with human participants or animals performed by any of the authors.
- **Informed Consent:** Not applicable.
- **Data Availability:** The dataset used in this study (Alibaba Cluster Trace 2017) is publicly available.
- **Code Availability:** The code used for this study is available from the corresponding author upon reasonable request.
- **Author Contributions:** All authors contributed equally to the conception, design, implementation, and writing of this study. All authors read and approved the final manuscript.

References

1. Lim B, Arik SO, Loeff N, Pfister T (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 37(4):1748–1764
2. Buyya R, Garg SK, Calheiros RN (2014) Auto-scaling techniques in cloud computing: Issues and research directions. Future Gener Comput Syst 33:1–20
3. Iqbal MS, Islam MS, Amin MR, Rahman MA (2022) Forecasting workload in cloud computing: Towards uncertainty-aware predictions and transfer learning. J Cloud Comput 11(1):1–21
4. Zhang Y, Wen J, Fang Y (2023) Multi-task learning and temporal-fusion-transformer-based forecasting of building power consumption. Energy Build 278:112628
5. Kertesz A, Kecskemeti G, Nemeth Z (2022) Enhancing machine learning-based autoscaling for cloud resource orchestration. IEEE Trans Cloud Comput 10(4):2385–2398

6. Hasan MK, Islam S, Kim J (2023) Optimizing multi-time series forecasting for enhanced cloud resource utilization based on machine learning. IEEE Access 11:12045–12060
7. Kim YM, Song S, Koo BM, Son J, Lee Y, Baek JG (2024) Enhancing long-term cloud workload forecasting framework: Anomaly handling and ensemble learning in multivariate time series. IEEE Trans Cloud Comput 12(2):789–804
8. Patil SR, Sahoo SK (2021) Deep CNN and LSTM approaches for efficient workload prediction in cloud environment. J Cloud Comput 10(1):1–18
9. Hossain MS, Bari MF, Buyya R (2020) LSTM-enabled dynamic Stackelberg game theoretic method for resource allocation in the cloud. Future Gener Comput Syst 113:479–492
10. Verma A, Kaushal S (2019) Enhancing cloud computing resource allocation with LSTM-based predictive modeling. Simul Model Pract Theory 94:31–45
11. Jeong B, Baek S, Park S, Jeon J, Jeong YS (2023) Stable and efficient resource management using deep neural network on cloud computing. Neurocomputing 521:99–112
12. Thonglek K, Ichikawa K, Takahashi K, Iida H, Nakasan C (2019) Improving resource utilization in data centers using an LSTM-based prediction model. In: Proc. IEEE Int. Conf. Cluster Comput., pp. 1–8
13. Ouhame S, Hadi Y, Ullah A (2021) An efficient forecasting approach for resource utilization in cloud data centers using CNN–LSTM model. Neural Comput Appl 33(16):10043–10055
14. Zi Y (2024) Time-series load prediction for cloud resource allocation using recurrent neural networks. J Comput Technol Softw 3(7)
15. Valarmathi K, Raja S (2021) Resource utilization prediction technique in cloud using knowledge-based ensemble random forest with LSTM model. Concur Eng 29(4):396–404
16. Yuan H, Bi J, Li S, Zhang J, Zhou M (2024) An improved LSTM-based prediction approach for resources and workload in large-scale data centers. IEEE Internet Things J
17. Paripati LK (2023) Optimizing resource allocation and scalability in cloud-based machine learning models. Migration Lett 20(S12):1823–1832
18. Bi J, Ma H, Yuan H, Zhang J (2023) Accurate prediction of workloads and resources with multi-head attention and hybrid LSTM for cloud data centers. IEEE Trans Sustain Comput 8(3):375–384
19. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Adv Neural Inf Process Syst, 30
20. Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proc. AAAI Conf. Artif. Intell. 35(12), 11106–11115
21. Wu H, Xu J, Wang J, Long M (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Adv Neural Inf Process Syst, 34
22. Nie Y, Nguyen NH, Sinthong P, Kalagnanam J (2023) A time series is worth 64 words: Long-term forecasting with transformers. In: Int Conf Learn Represent

23. Oreshkin BN, Carpov D, Chapados N, Bengio Y (2020) N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In: Int Conf Learn Represent
24. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
25. Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4)
26. Lorido-Botran T, Miguel-Alonso J, Lozano JA (2014) A review of auto-scaling techniques for elastic applications in cloud environments. J Grid Comput 12(4):559–592
27. Mao M, Humphrey M (2012) A performance study on the VM startup time in the cloud. In: Proc. IEEE Int. Conf. Cloud Comput., pp. 423–430
28. Herbst NR, Kounev S, Reussner R (2013) Elasticity in cloud computing: What it is, and what it is not. In: Int Conf Auton Comput, pp. 23–27
29. Chen X, Liu H, Chen X (2019) Machine learning based workload prediction in cloud computing. IEEE Access 7:125986–125997
30. Xu J, Fortes JAB (2010) Multi-objective virtual machine placement in virtualized data center environments. In: Proc. IEEE/ACM Int. Symp. Cluster Cloud Grid Comput., pp. 179–188

Figures

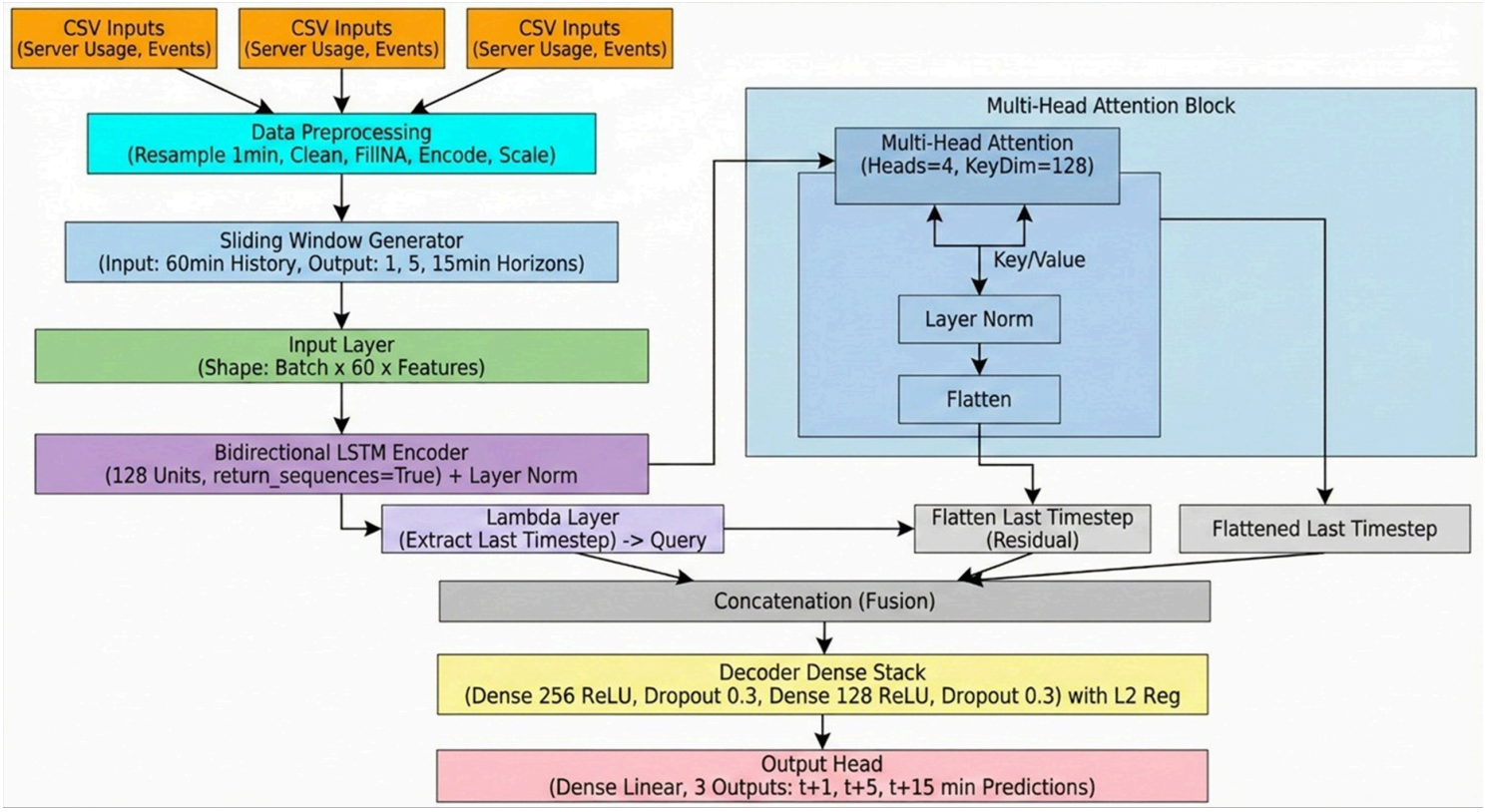


Figure 1

Architecture of the proposed Temporal Fusion Transformer (TFT) model for multi-horizon cloud resource demand prediction

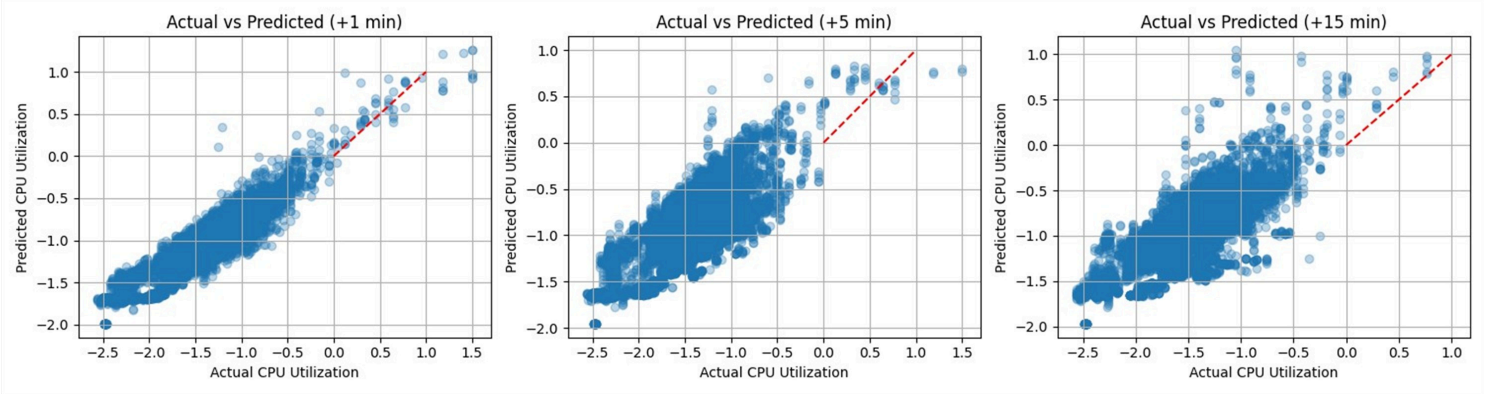


Figure 2

Comparison between actual and predicted CPU utilization values for the proposed Temporal Fusion Transformer (TFT) model

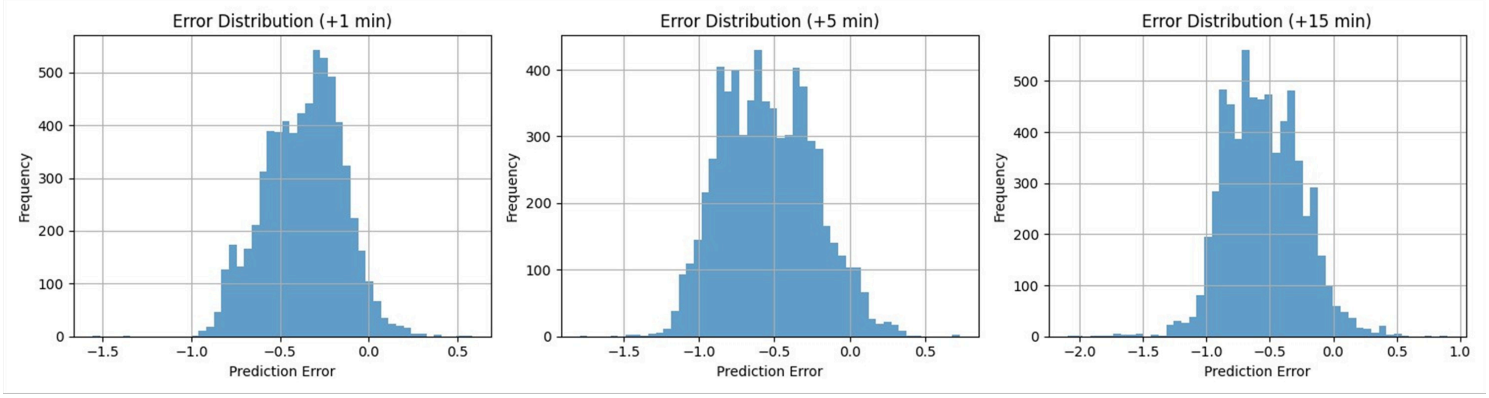


Figure 3

Distribution of prediction errors across different forecasting horizons for CPU utilization using the proposed Temporal Fusion Transformer (TFT) model

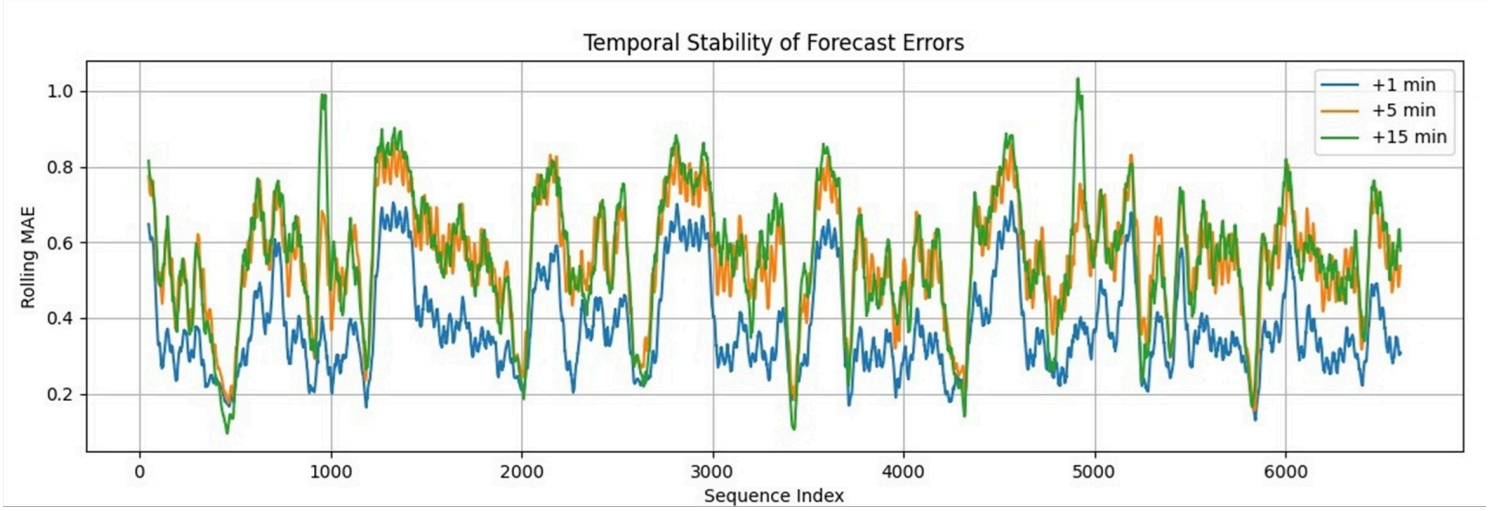


Figure 4

Temporal stability of rolling mean absolute error (MAE) across different forecasting horizons for the proposed Temporal Fusion Transformer (TFT) model

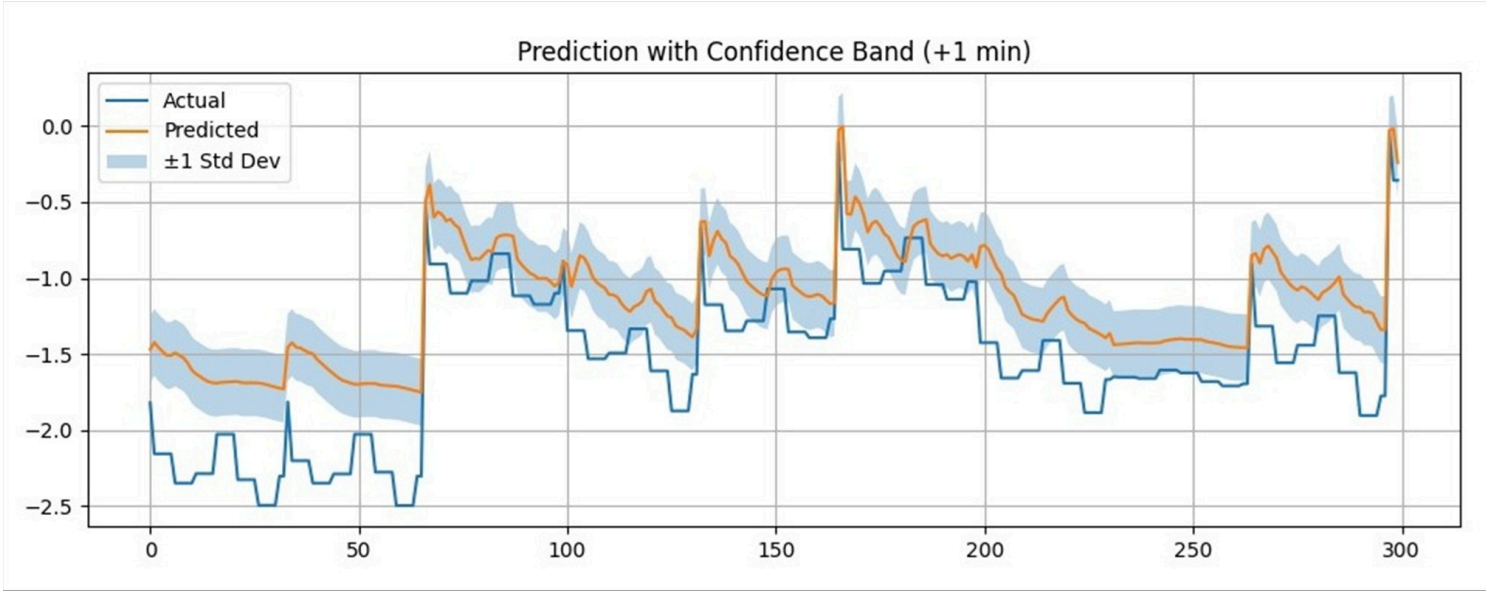


Figure 5

CPU utilization forecast with confidence interval for the +1 minute prediction horizon using the proposed Temporal Fusion Transformer (TFT) model

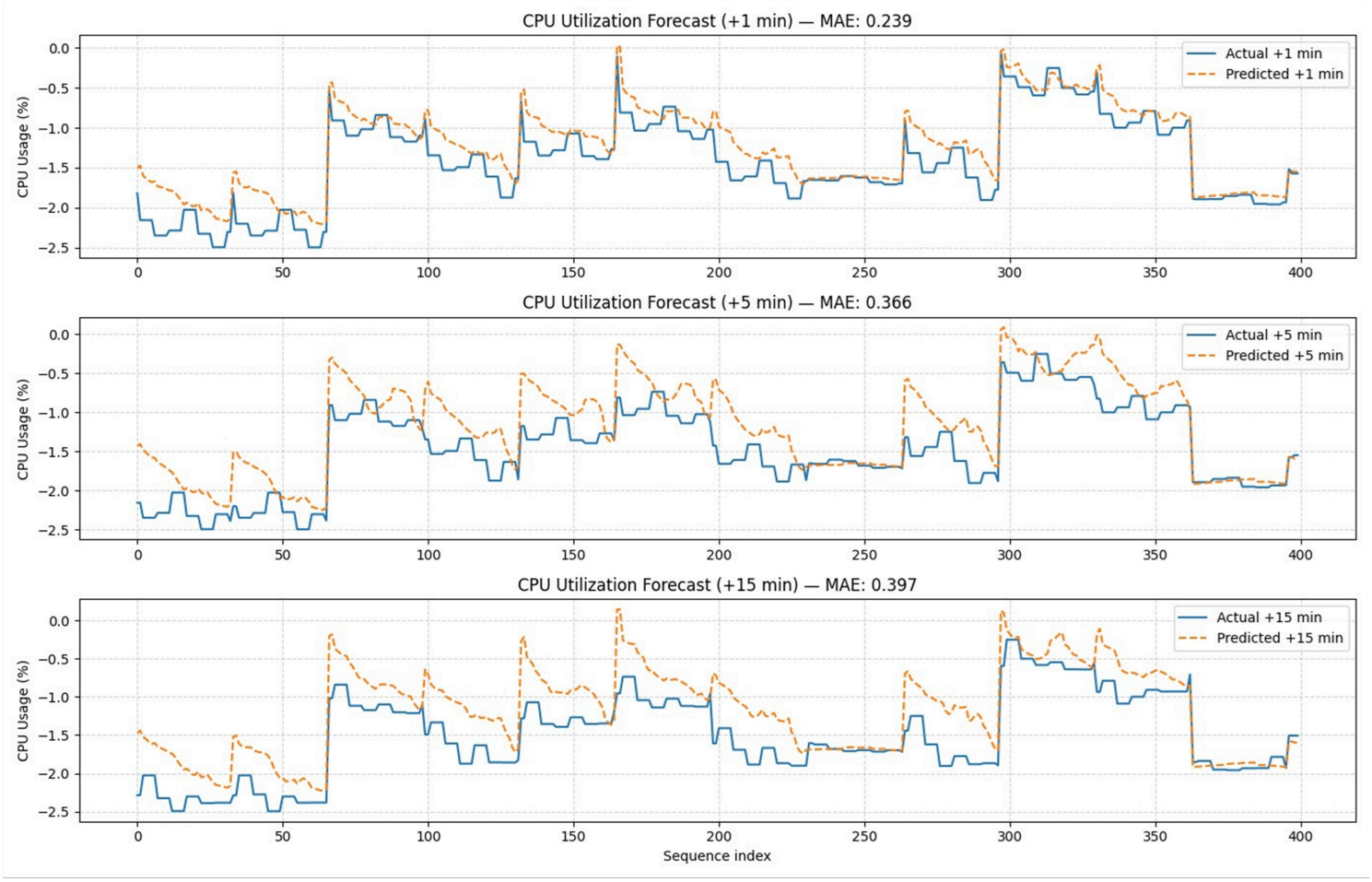


Figure 6

Comparison of actual and predicted CPU utilization across multiple forecasting horizons using the proposed Temporal Fusion Transformer (TFT) model