Journal of Artificial Intelligence and Technology, (Ahead of Print)
https://doi.org/10.37965/jait.2026.1852

ISTP
RESEARCH ARTICLE

A data-Driven Approach To Using Artificial Intelligence To Improve Network Communication Efficiency

YuSheng Jiang, Muhan Li, and Ni Liang
School of Information and Future Technology, The Open University of Shaanxi, Shaanxi, Xi'an; 710068, China

(Received 10 August 2026; Revised 09 October 2026; Accepted 19 October 2026; Published online 25 November 2026)

Abstract: The rapid growth of data-driven communication networks has increased the need for accurate network traffic analysis and throughput prediction to support efficient resource utilization and intelligent network management. However, conventional prediction methods often struggle to capture complex temporal dependencies, nonlinear variations, and dynamic traffic patterns, resulting in limited forecasting accuracy and generalization. This study proposes a deep learning-based network throughput prediction framework using the CESNET-TimeSeries24 dataset, which contains large-scale, regularly sampled network traffic statistics. The framework incorporates data preprocessing, including missing-value handling, outlier treatment, feature normalization, and sliding-window transformation for effective temporal modeling. A Bidirectional Long Short-Term Memory (BiLSTM) network is employed to learn both short- and long-term traffic dependencies from forward and backward temporal sequences. Experimental results demonstrate strong predictive performance, achieving an mean absolute error (MAE) of 0.0306, mean squared error (MSE) of 0.0018, and root mean squared error (RMSE) of 0.0429. The findings confirm that the proposed framework provides accurate and consistent throughput predictions. These predictions may potentially support intelligent traffic analysis, efficient resource allocation, and network performance optimization in modern high-speed communication networks; however, these downstream network management benefits are not directly evaluated in the present study.

Keywords: BiLSTM; deep learning; network traffic prediction; throughput prediction; time-series forecasting

I. INTRODUCTION

The efficiency of network communication is a key part that supports modern digital systems, allowing everything from remote work to live video streaming and large sensor networks [1]. The field of communication networks has experienced continuous development during the last three decades because of growing data requirements and improved transmission technologies and the rising use of network connections in homes and businesses [2]. The research area has experienced continuous growth because academic and industrial organizations have devoted resources to studying and improving network performance [3]. Researchers from previous studies have established traffic growth trends and protocol development processes and capacity management methods that apply to multiple network types [4]. The ongoing research demonstrates how efficient network communication functions as the essential factor that enables systems to deliver dependable, high-quality performance across their connected network segments [5].

Research in network traffic analysis and prediction has developed multiple methodological approaches during its different research stages [6]. The initial research used traditional statistical methods to study network traffic patterns and identify network flow irregularities [7]. Machine learning (ML) methods created models that learned to understand traffic data through nonlinear connections [8]. Deep learning (DL) systems that employ recurrent neural networks with their various types for processing sequential data have achieved substantial progress in predicting future traffic patterns during the past three years [9]. The research studies have provided important knowledge about how to create models that simulate traffic flow changes, and they have developed strategies for dynamic routing that enhance communication efficiency [10,11]. The research work establishes a basis for developing stronger data-based forecasting systems that network researchers can use in their studies. This study introduces a framework that uses data to create an artificial intelligence system that predicts network throughput for its purposes of measuring communication efficiency. The study falls within the scope of ML and big data processing by applying DL to large-scale time-series network traffic data for intelligent throughput prediction.

The proposed framework effectively addresses the shortcomings of existing approaches by incorporating a Bidirectional Long Short-Term Memory (BiLSTM)-based AI model capable of capturing both short-term fluctuations and long-term dependencies in network traffic. Leveraging large-scale datasets such as CESNET-TimeSeries24, the framework delivers highly accurate predictions while maintaining low computational overhead. Unlike conventional methods, it provides rapid and dependable throughput forecasts, enabling more efficient resource management and proactive network optimization. The practical contribution of this study lies in integrating an established BiLSTM architecture with a structured data-processing pipeline for network throughput prediction using the CESNET-TimeSeries24 dataset. In addition, the framework is designed to adapt dynamically to evolving network conditions, allowing operators to respond quickly to unexpected traffic surges or fluctuations. By overcoming the limitations of prior techniques, this approach not only improves network throughput prediction but also establishes a practical foundation for next-generation intelligent network management systems. Ultimately, it supports more resilient,

Corresponding author YuSheng Jiang (email: 101220893@ousn.edu.cn).

© The Author(s) 2026. This is an open access article published under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

efficient, and responsive networks capable of handling complex and unpredictable traffic patterns, paving the way for smarter and more reliable communication infrastructures.

The novelty of this study lies in integrating an established BiLSTM model with a structured preprocessing and feature-selection pipeline for large-scale network throughput prediction using CESNET-TimeSeries24. The framework combines mean imputation, Z-score-based outlier detection, min–max normalization, and correlation-based feature selection to preprocess the network traffic data before model training.

The main contributions of this study are as follows:

- Applies an established BiLSTM framework to the CESNET-TimeSeries24 dataset for network throughput prediction, with the evaluation focused on predictive accuracy rather than direct measurement of downstream communication-efficiency outcomes.
- Implements a structured data-processing pipeline, including mean imputation, Z-score-based outlier detection, min–max normalization, correlation-based feature selection, and sliding-window sequence generation.
- Uses the processed network traffic sequences to train the BiLSTM model to capture temporal dependencies for throughput forecasting.
- Evaluates the framework quantitatively using mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R^2 to assess its network throughput prediction performance.
- Clarifies that the contribution of this study focuses primarily on applying and evaluating an established BiLSTM framework rather than developing a new BiLSTM architecture.

The paper is organized as follows. Section I introduces the problem, explains the motivation behind this study, and highlights the key contributions. The existing research is summarized in Section II, while the research gaps that this work intends to address are identified. The proposed methodology is explained through its detailed description in Section III. The fourth section displays experimental results together with an extensive analysis of performance metrics. The paper ends in Section V, which presents research paths that can enhance network communication efficiency for upcoming studies.

A. PROBLEM STATEMENT

Despite significant advancements in AI-based network traffic prediction, integrating these techniques with communication efficiency mechanisms continues to face major challenges. One of the primary issues is developing models capable of accurately capturing complex spatiotemporal patterns in diverse network environments, where dynamic user behavior, fluctuating traffic loads, and varying environmental conditions significantly impact network performance [12]. In software-defined networks (SDNs), critical problems such as congestion from sudden traffic surges and uneven load distribution across programmable infrastructure remain prevalent [13]. Similarly, vehicle-to-everything (V2X) networks operate under tight resource constraints, requiring precise traffic forecasts to maintain operational performance, yet many existing DL models struggle due to computational limitations, poor scalability, and limited adaptability to changing network conditions [14]. Moreover, current approaches often fail to strike a balance between prediction accuracy and processing efficiency, producing solutions that are either slow or insufficiently precise. These gaps underscore the need for a unified framework that delivers reliable, adaptive, and efficient throughput predictions. To address these challenges, this study proposes an AI-based framework that leverages sequential learning techniques to model complex traffic patterns from large-scale datasets. By combining accurate prediction with efficient resource utilization, the framework enhances communication performance, supports dynamic network adaptation, and provides a practical foundation for intelligent, next-generation network management systems capable of handling unpredictable traffic scenarios effectively.

II. LITERATURE REVIEW

The growing demand for effective network communication systems has increased the need for intelligent approaches to network monitoring, traffic prediction, resource management, and performance optimization. Cui [15] showed that AI functions as an essential element throughout all components of sixth-generation networks because it enables better resource distribution and system performance and development, from basic functions to complete AI solutions used in virtual reality and industrial settings. The study conducted by El-Hajj [16] proved that AI has practical uses for predicting traffic patterns, managing network load, detecting security threats, and operating self-managing networks, which increases both security and operational efficiency of dense networks that use fifth- and sixth-generation technology and IoT devices. The study conducted by Tok [17] investigated how AI and ML technology function within modern communication systems, which include advanced wireless systems combining multiple-input multiple-output (MIMO), reconfigurable intelligent surface (RIS), and integrated sensing and communication (ISAC) for enhanced communication and sensing capabilities, and showed that these technologies can increase capacity while decreasing latency and delivering better performance. Al-Otaibi [18] created a mobile ad hoc network system that uses deep reinforcement learning (DRL) to manage network routing and network slicing functions to achieve major enhancements in both packet delivery and network throughput and end-to-end delay. The research conducted by Khan [19] used digital twin simulations together with Low Energy Adaptive Clustering Hierarchy energy-efficient protocols to achieve better reliability and energy performance in Internet of Things networks. All of these studies demonstrated that AI technology can boost network performance through their research results, but their research suffered from two main limitations: they could not function at large scale, and they failed to address various network operational settings that exist in heterogeneous environmental conditions.

Researchers developed advanced data-driven optimization techniques to address complex challenges arising in fifth- and sixth-generation communication networks.

Aziz [20] demonstrated a dual-step transfer learning framework integrating RNN and BiLSTM models for multivariate spatio-temporal cellular traffic prediction. The study achieved improved forecasting accuracy with reduced training time and computational requirements.

Saleh [21] evaluated a double deep Q-network with prioritized experience replay (DDQN-PER) and Pointer Network (PtrNet)-LSTM-based resource allocation framework for dynamic 5G network slicing environments. The framework achieved enhanced spectrum efficiency, improved user connectivity, and reliable communication performance under varying network conditions. Aleran [22] demonstrated that AI techniques enable energy conservation in IoT networks. The architecture would consist of three

layers: data collection, energy management, and control. Optimization, automation, and interoperability were identified as key challenges. Kamal [23] proposed the hHGGO_XCovNet framework, which uses DL techniques along with hybrid optimization techniques to improve resource allocation, energy efficiency, throughput, and sum rate in multi-input multi-output-enabled networks. Saoud *et al.* [24] investigated techniques for integrating mobility and handover management in fifth- and sixth-generation networks. The study showed that AI integration can improve quality of service (QoS), reduce latency, and enhance energy utilization. All these studies show that AI has immense potential in transforming communication networks. However, it is still constrained by the lack of joint optimization between heterogeneous layers of networks.

The rapid evolution of wireless networks has emphasized the requirement to utilize advanced data-driven techniques to ensure effective communication. For instance, the research by Konatam [25] proved that the application of AI in traffic prediction and load balancing improved resource allocation, reduced congestion, and ensured reliability in fifth- and sixth-generation networks. Another study by Jiang *et al.* [26] focused on the application of graph neural network (GNN) in routing optimization in unmanned aerial vehicle networks, satellite networks, and fifth-generation networks. The results proved that the application of GNN improved scalability and flexibility in network communication. Another study by Duan *et al.* [27] proposed an AI-agent communication network paradigm, focusing on the application of AI to the development of autonomous communication networks. This research proved that the application of AI in communication networks improved the creation of an autonomous communication network that addressed the challenges associated with multimodal interactions and privacy preservation. Another study by Omheni *et al.* [28] examined the application of AI in network functions virtualization (NFV) and SDN technologies to improve operational efficiency, network flexibility, and predictive management capabilities in modern communication systems. Banaie *et al.* [29] proposed an AI-based modular, serverless framework for creating energy-efficient networks. The research demonstrated that AI-based network function deployment can improve scalability and reduce power consumption in communication networks. All these studies proved that AI has the potential to improve communication efficiency in networks but is still limited by the inability to ensure interoperability between communication layers.

III. METHODOLOGY

The proposed framework is an AI-based system designed to provide accurate network throughput predictions while enabling users to monitor and manage network behavior effectively in dynamic environments. The workflow begins with data acquisition from the CESNET-TimeSeries24 dataset, which contains extensive network traffic records and forms the foundation for the structured operation of the model. After data collection, a thorough preprocessing step is carried out to handle missing values, remove outliers, and normalize the data, ensuring consistency and reliability for subsequent analysis. Feature extraction follows, where correlation-based selection identifies the most relevant features that significantly influence network traffic patterns. Sequence generation is then applied using a sliding-window approach, which enables the model to capture temporal dependencies across the dataset. The BiLSTM model serves as the core prediction component, selected for its capability to learn both forward and backward temporal relationships in sequential data, improving overall prediction accuracy. To evaluate model performance, metrics such as MAE, RMSE, and R^2 are used, providing quantitative measures of prediction reliability and robustness. By integrating data acquisition, preprocessing, feature extraction, and sequence learning in a systematic manner, the proposed framework ensures precise and reliable throughput forecasts. This structured methodology not only enhances communication efficiency but also provides actionable insights for dynamic network management, as illustrated in Fig. 1.

A. DATASET DESCRIPTION

The CESNET-TimeSeries24 dataset [30], collected over a span of 40 weeks from CESNET3, provides comprehensive network traffic statistics for more than 275,000 IP addresses. These include a diverse range of devices such as office workstations, servers, routers, and gaming consoles, capturing the complexity of modern network environments. The dataset records traffic statistics at multiple timescales, including 10-minute, 1-hour, and 24-hour intervals, encompassing metrics such as flow counts, packets and bytes sent, unique destinations, protocol ratios, direction ratios, average flow durations, and time-to-live values. This multi-timescale and multi-metric approach allows the dataset to support both short-term and long-term analysis, making it particularly suitable for network throughput prediction and anomaly detection tasks. The time-series data are organized not only at the level of individual IP addresses but also aggregated across institutions and subnets, enabling hierarchical analysis of network behavior. This structure facilitates detailed examination of traffic patterns and dependencies across different network entities. In this study, the CESNET-TimeSeries24 dataset serves as the foundation for training and evaluating the proposed BiLSTM-based AI framework. Its comprehensive coverage and hierarchical organization provide the necessary granularity and scale to model complex temporal patterns accurately, making it highly suitable for assessing network performance, understanding traffic dynamics, and improving communication efficiency in large-scale network environments.

B. DATASET PREPROCESSING

It is essential to do proper dataset preprocessing to guarantee the reliability, consistency, and quality of the input data prior to feeding it to ML models. The preprocessing process in this research involved the processing of missing values, outliers, and scaling of features, hence creating a coherent and machine-readable dataset. All these measures minimized bias and variance and improved the predictive ability and stability of the models.

The systematic analysis of missing values was conducted to guarantee the completeness of the dataset and means of reliable training of the model. In the case of numerical features, mean

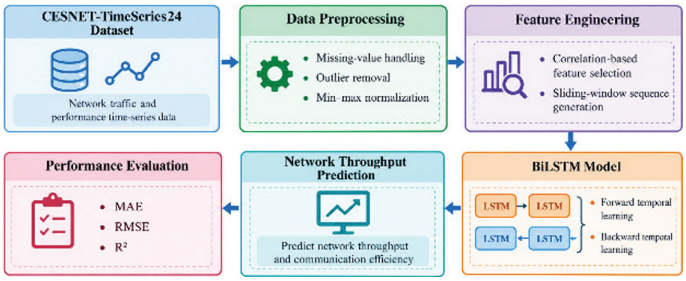


Fig. 1. Proposed AI framework for network throughput prediction and communication efficiency.

imputation was used, and the missing items were identified by the average of the available data. The common imputation is determined as Eqn. (1):

$$x_i = \frac{1}{N}\sum_{j=1}^{N} x_j \tag{1}$$

where x_i represents the value imputed for a missing data point, N denotes the number of non-missing samples available for that particular feature, and x_j corresponds to the observed values of that feature. This approach calculates the mean of all valid observations to fill in missing entries, ensuring consistency in the dataset. By imputing missing values in this way, the method preserves the statistical properties of the data and provides a reliable foundation for subsequent analysis and modeling tasks.

Outliers were identified based on Z-score outliers and eliminated since the values that are outliers are very different to the mean of the feature. This move will help the model to avoid bias due to extreme values of the traffic. The change is specified as in Eqn. (2):

$$z_i = \frac{x_i - \mu}{\sigma}; \mid z_i \mid \leq 3 \tag{2}$$

where x_i is the original feature value, and μ and σ represent the mean and standard deviation of that feature, respectively. In this approach, the Z-score standardization is applied to normalize the data, ensuring that each feature contributes proportionally to the analysis. Values with $|z_i| > 3$ are considered outliers and are treated accordingly, either by removal or adjustment, to prevent extreme values from negatively affecting the performance of subsequent modeling steps such as feature extraction and prediction.

Feature scaling was applied to normalize continuous variables and ensure consistent ranges across all features. Min–max normalization was used to transform feature values into the range [0,1], as defined by Eqn. (3):

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

where x_i' represents the scaled feature value, x_i is the original feature value, and $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values of that feature, respectively. This normalization technique ensures that all feature values are rescaled within a consistent range, typically between 0 and 1, which improves the stability and convergence of the learning process. By applying this scaling, the model can handle features with different ranges more effectively, leading to better performance in tasks such as network traffic prediction.

After completing the preprocessing steps, the prepared network traffic time-series dataset was divided into training, validation, and testing subsets to support effective model training and unbiased performance evaluation. In this study, a data split ratio of 70:15:15 was adopted. The training subset was used to learn temporal traffic patterns, while the validation subset assisted in hyperparameter tuning and overfitting prevention. The testing subset was reserved for the final evaluation of the proposed BiLSTM model on previously unseen traffic observations. Since the dataset consists of sequential time-dependent measurements, the chronological order of samples was preserved during the splitting process to avoid data leakage and maintain realistic prediction conditions.

C. FEATURE EXTRACTION

Efficient feature extraction converts raw network traffic data to ordered inputs to the time-series model so that they are learned to capture significant temporal relationships. The pipeline that has been used in the study consists of correlation analysis, sequential input construction based on a sliding window, and dataset splitting to optimize the prediction accuracy and generalization.

Correlation analysis was also used so that the features that are most useful in the prediction of network traffic could be identified. The linear relationship between each of the inputs xi and the target variable y was quantified by the Pearson correlation coefficient. It is expressed in Eqn. (4):

$$r_{x_i,y} = \frac{\sum_{j=1}^{N}(x_{ij} - \bar{x_1})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{N}(x_{ij} - \bar{x_1})^2 \sum_{j=1}^{N}(y_j - \bar{y})^2}} \tag{4}$$

where $r_{x_i,y}$ is the correlation coefficient between feature x_i and the target y, x_{ij} is the j^{th} sample of feature x_i, y_j represents j^{th} target value, $\bar{x_1}$ and $\bar{y}$ are the mean value of the feature and the target, respectively, and N is the number of samples. Highly correlated features were modeled, and weakly correlated or redundant features were filtered to avoid noise and streamline modeling. A sliding-window mechanism was implemented to represent temporal dependencies of traffic on a network.

Each input sequence X_t is formed from w consecutive time steps from the past, capturing historical trends and patterns in the data. The target $y_{(t+1)}$ represents the value at the next time step that the model aims to predict. This relationship is formally defined in Eqn. (5), enabling the model to learn temporal dependencies for accurate forecasting:

$$X_t = [x_{t-w+1}, x_{t-w+2}, \ldots, x_t],\ y_{t+1} = x_{t+1} \tag{5}$$

where X_t is the previous w time steps of the input features, y_{t+1} is the subsequent target traffic value, and w is the window size. Models that can be used to effectively learn the temporal patterns and dependencies in network traffic can be constructed using this sequential construction, including recurrent neural networks and gradient boosting algorithms.

Once the sequential samples were prepared, the dataset was divided into training, validation, and testing subsets to prevent biased evaluation. A standard 70:15:15 split was applied, ensuring that the model could learn effectively from the training data, validate its performance during training, and be fairly tested on unseen data. This process is formally represented in Eqn. (6):

$$N_{\text{train}} = 0.7 \times N, N_{\text{val}} = 0.15 \times N, N_{\text{test}} = 0.15 \times N \tag{6}$$

where N is the total number of sample count of the sequential samples and N_{train}, N_{val}, and N_{test} represent the sample count of a training set, validation set, and test set, respectively. Such a divide provides adequate learning data, appropriate hyperparameter optimization, and valid analysis of unknown data, which eventually enhances the generalization and the strength of the models.

D. MODEL DEVELOPMENT

The model development phase establishes a predictive framework centered on a BiLSTM network, specifically designed to forecast network throughput with high precision. Prior to modeling, network traffic attributes are preprocessed and structured into sequential time series that capture the evolution of communication behavior over consecutive time intervals. These multidimensional traffic features represent critical characteristics, including packet transmission rates, flow durations, protocol distributions, and historical throughput trends, providing a comprehensive view of network dynamics. The sequential input data is then fed into the BiLSTM architecture, which is capable of modeling both forward

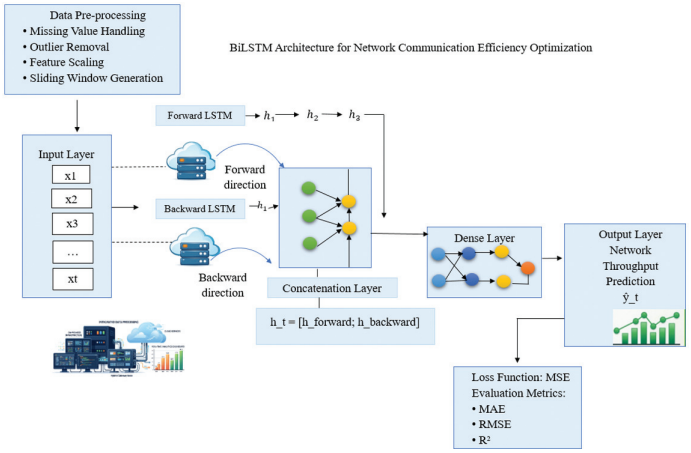


Fig. 2. BiLSTM directional architecture for network throughput prediction.

and backward temporal dependencies within the traffic stream. By simultaneously considering past and future observations, the BiLSTM effectively learns contextual relationships across the sequence, enhancing its ability to represent complex, dynamic traffic patterns. This bidirectional learning capability allows the model to capture subtle temporal variations and interdependencies that traditional unidirectional models may overlook. As illustrated in Fig. 2, the framework leverages these strengths to improve throughput prediction accuracy, offering a robust approach for understanding and managing the dynamic behavior of modern communication networks.

A forward LSTM layer processes the sequential traffic inputs in chronological order to capture temporal dependencies from earlier network states to later states. Each LSTM unit maintains an internal memory state that enables the model to retain relevant information across time steps while regulating information flow through specialized gating mechanisms, as shown in Eqn. (7):

$$h_t^f = LSTM_f(x_t, h_{t-1}^f) \tag{7}$$

where x_t represents the input feature vector at time step t, h_t^f denotes the forward hidden state generated by the LSTM cell, and h_{t-1}^f corresponds to the hidden state from the previous time step in the forward sequence. By preserving relevant information from past states, the forward LSTM enables the network to model temporal patterns in traffic data effectively, making it capable of learning complex sequential relationships and improving the accuracy of network throughput predictions.

The backward LSTM layer simultaneously processes the input sequence in the reverse temporal direction to capture contextual dependencies from future observations. This bidirectional learning mechanism allows the network to obtain a more comprehensive representation of temporal traffic dynamics as shown in Eqn. (8):

$$h_t^b = LSTM_b(x_t, h_{t+1}) \tag{8}$$

where h_t^b denotes the hidden state produced by the backward LSTM unit at time step t, and h_{t+1}^b represents the hidden state from the next time step in the reverse sequence. During the backward pass, the LSTM processes the input sequence in reverse order, allowing the model to capture future context relative to the current step. By integrating both forward and backward hidden states, the BiLSTM effectively models temporal dependencies in both directions, enhancing sequence learning and improving prediction performance in tasks such as network traffic forecasting.

The hidden representations obtained from the forward and backward LSTM layers are merged to form a unified feature representation that integrates bidirectional temporal information. This concatenated representation enables the model to capture both historical and future traffic dependencies simultaneously as is shown in Eqn. (9):

$$h_t = [h_t^f; h_t^b] \tag{9}$$

where h_t denotes the combined hidden representation obtained by concatenating the forward hidden state h_t^f and the backward hidden state h_t^b. This concatenation allows the BiLSTM to capture information from both past and future contexts simultaneously, providing a richer and more informative representation of the sequence at time step t. By integrating insights from both directions, the model can better understand temporal dependencies, enhancing its ability to make accurate predictions in sequential tasks such as network traffic forecasting.

The concatenated hidden representation is then supplied to a fully connected prediction layer, which transforms the learned temporal features into a continuous output corresponding to the predicted network throughput value, as shown in Eqn. (10):

$$\hat{y}_t = W_o h_t + b_o \tag{10}$$

where $\hat{y}_t$ represents the predicted network throughput at time step t, W_o denotes the output weight matrix, b_o is the bias term, and h_t corresponds to the concatenated hidden representation obtained from the BiLSTM layer. This formulation maps the learned temporal features from the BiLSTM to the output space, generating the predicted throughput. By combining forward and backward hidden states, the model leverages full contextual information, enabling more accurate and reliable predictions of network traffic patterns.

The training objective aims to minimize the deviation between predicted and actual throughput values by optimizing the model parameters through a regression loss function. The prediction error is quantified using the mean squared error across all training samples as is shown in Eqn. (11):

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{11}$$

where L denotes the prediction loss, N represents the total number of training instances, y_i is the actual throughput value, and $\hat{y}_i$ is the predicted throughput generated by the model. This loss function, known as the MSE, measures the average squared difference between the predicted and actual values. Minimizing L during training ensures that the model produces more accurate throughput predictions, effectively improving network traffic forecasting performance.

The optimization process iteratively updates the model parameters to minimize the prediction loss and ensure convergence toward an optimal configuration capable of accurately capturing complex temporal traffic patterns, as is shown in Eqn. (12):

$$\theta^* = \arg\min_{\theta} L \tag{12}$$

where θ^* represents the optimized set of parameters for the BiLSTM model, and the minimization operation identifies the specific parameter configuration that produces the lowest prediction error over the training dataset. This optimization process ensures that the model learns the most effective weights and biases to accurately capture temporal patterns in the input sequences. By finding θ^*, the BiLSTM achieves improved prediction performance, enabling reliable forecasting and efficient modeling of

Algorithm 1. BiLSTM-Based Network Throughput Prediction Framework

Input: CESNET-TimeSeries24 dataset containing packet counts, byte counts, flow statistics, protocol ratios, and historical throughput.

Output: Predicted network throughput values and performance metrics.

Begin

1. Load the CESNET-TimeSeries24 dataset.
2. Perform data preprocessing, including missing-value handling, outlier elimination, and feature scaling.
3. Construct traffic feature vectors representing network statistics at time t.
4. Perform correlation analysis to identify relevant prediction features.
5. Generate time-series sequences using a sliding-window mechanism.
6. Split the dataset into training, validation, and testing sets.
7. Initialize the BiLSTM model.
8. Define the forward and backward LSTM layers.
9. Train the BiLSTM model using the training sequences.
10. Generate predicted network throughput values.
11. Evaluate model performance using MAE, MSE, RMSE, and R^2.
12. Return the predicted throughput values and evaluation results.

End

sequential data such as network traffic. The methodology aligns with network intelligence by using a BiLSTM-based framework to learn temporal patterns in network traffic and predict future network throughput.

E. SYSTEM-LEVEL EVALUATION PROCEDURE

The system-level evaluation considers three representative network load conditions: low, medium, and high load, to examine network behaviour under varying traffic intensities. The conventional network management approach is considered the baseline, whereas the proposed BiLSTM-based prediction-assisted approach uses the predicted throughput information to support network management decisions. For each load condition, average throughput, average latency, packet loss, and resource utilization are measured. Throughput improvement, latency reduction, and packet-loss reduction are subsequently calculated relative to the corresponding baseline measurements. This evaluation complements the regression-based throughput prediction analysis by examining the association between predicted throughput information and system-level communication performance.

IV. RESULT

The primary aim of this study is to evaluate the performance of the proposed AI-based framework for network throughput prediction using the CESNET-TimeSeries24 dataset. Extensive experiments were conducted to assess the ability of the BiLSTM model to capture temporal patterns and complex dependencies in sequential network data. Regression-based performance metrics were used to evaluate prediction accuracy and reliability. The model achieved a MAE of 0.0306, MSE of 0.0018, and RMSE of 0.0429, demonstrating accurate and consistent throughput prediction. Although the dataset provides comprehensive time-series network traffic statistics, it does not contain direct system-level information for evaluating resource allocation, network optimization, QoS, latency, congestion, or packet loss. Therefore, the present experiments focus on throughput prediction accuracy, while the potential contribution of the predicted throughput information to broader communication-efficiency-oriented network management is not directly evaluated.

A. SYSTEM SPECIFICATION

The experimental research is based on the CESNET-TimeSeries24 dataset that consists of large-scale network traffic statistics of monitoring infrastructure. The dataset has time-series data, which demonstrate the continuous network throughput and traffic patterns over the period it was observed. The selected network traffic conditions represent realistic operating scenarios for evaluating the throughput prediction capabilities of the proposed AI model and its potential application to communication-efficiency evaluation. The Python language is used to implement the suggested framework. The CESNET-TimeSeries24 dataset provides large-scale time-series network traffic statistics suitable for throughput prediction. However, it does not directly provide system-level information related to resource allocation, network optimization, QoS, latency, congestion, or packet loss. The system-level communication-efficiency evaluation described in Section III.E uses the predicted throughput values to examine network performance under different load conditions. The authors employ NumPy, Pandas, and Scikit-learn to perform data preprocessing and feature engineering and construct the BiLSTM prediction model using TensorFlow/Keras. Matplotlib and Seaborn help researchers to visualize the network traffic patterns and prediction results. The data are partitioned into the training, validation, and testing dataset in the proportions of 70:15:15 to ensure effective evaluation of the model Table I.

1). EXPERIMENTAL CONFIGURATION. The experiments were carried out using Python 3.9 along with essential libraries, including NumPy, Pandas, scikit-learn, and TensorFlow/Keras, as well as Matplotlib and Seaborn for data visualization. The development environment included Jupyter Notebook and VS Code, running on both Windows 10 and Ubuntu 20.04 systems. Furthermore, Google Colab's GPU can be used as an alternative for efficient model training and handling computationally intensive tasks during experimentation.

B. PERFORMANCE EVALUATION

The performance of the developed framework, based on the BiLSTM approach, in predicting network throughput is evaluated in this section. The predicted network throughput is compared with the network traffic, and the performance of the model is evaluated for

Table I. Hyperparameter tuning configuration for proposed BiLSTM model

S. No.	Hyperparameter	Tested values	Selected value
1	Learning rate	0.01, 0.001, 0.0005	0.001
2	Batch size	16, 32, 64	32
3	BiLSTM units	32, 64, 128	64
4	Dropout rate	0.1, 0.2, 0.3	0.2
5	Optimizer	Adam, RMSprop	Adam
6	Number of epochs	50, 75, 100	75
7	Activation function	Tanh, ReLU	Tanh
8	Sliding window size	5, 10, 20	10

accuracy. To compare the performance of the model, regression performance measures, such as MAE, MSE, RMSE, and R^2, are used.

The predicted throughput values $\hat{y}_i$ generated by the proposed BiLSTM model find their initial comparison with actual throughput observations y_i. The absolute difference between predicted and actual values is computed for each observation and averaged to obtain the overall prediction error, as defined in Eqn. (13):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid \tag{13}$$

where y_i represents the actual network throughput value at the i^{th} observation and $\hat{y}_i$ denotes the corresponding predicted throughput value generated by the model. The term n indicates the total number of samples in the dataset, while i refers to the the index of each observation. MAE calculates the average absolute difference between the predicted and actual values, providing a straightforward measure of prediction accuracy. Lower MAE values indicate better model performance and more reliable throughput forecasting, which is critical for effective network management and resource allocation.

To further assess prediction performance, there is a squared difference between predicted and actual values of throughput that is calculated. This expression focuses more on the greater errors, which enables more sensitive determination of stability of prediction as in the form of Eqn. (14):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

where y_i represents the true throughput value, $\hat{y}_i$ denotes the predicted value, n is the total number of samples, and i is the observation index. MSE calculates the average squared difference between actual and predicted values, providing a clear indication of prediction accuracy. This metric is particularly useful for identifying large deviations, as squaring the errors amplifies the impact of significant prediction discrepancies. By using MSE, the model's performance can be quantitatively evaluated, helping ensure reliable and robust network throughput forecasts.

A square root of the mean squared error is calculated in order to put the prediction error in the same unit as the original throughput values. RMSE gives an understandable measure of the accuracy of prediction and it is stated in Eqn. (15):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

where y_i denotes the actual throughput value, $\hat{y}_i$ represents the predicted throughput, n is the total number of samples, and i refers to the observation index. RMSE measures the square root of the average squared differences between actual and predicted values, capturing both the magnitude and variance of prediction errors. A lower RMSE indicates that the predictions closely match the actual throughput, reflecting the model's accuracy and reliability in forecasting network performance across the dataset. This metric is widely used for evaluating regression-based prediction models.

The coefficient of determination is used to evaluate how well the predicted throughput values approximate the actual observations. The coefficient of determination (R^2) is used to evaluate how well the predicted throughput values approximate the actual observations. It represents the proportion of variance in the observed throughput values that is explained by the model, with higher R^2 values indicating better agreement between the predicted and actual values, as expressed in Eqn. (16):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

where y_i represents the actual throughput values, $\hat{y}_i$ denotes the predicted values, is the mean of the actual observations, n is the total number of samples, and i refers to the index of each observation. A high R^2 value indicates that the model effectively captures the underlying patterns in the data. It demonstrates strong predictive capability, showing that the model can accurately explain the variance in the observed throughput values and reliably forecast future network behavior.

Figure 3 provides a temporal analysis that can be used to demonstrate the predictive performance of the proposed model by comparing the actual throughput (blue solid line) and predicted throughput (red dashed line). The model effectively captures the inherent cyclic trends which make normalized throughput values to fluctuate between a minimum of 0.05 and the peak of 0.61. The two curves exhibit an extremely large level of overlap, which proves the fact that the model is capable of accurately predicting changes in network loads provided that it maintains its performance under real-world resource-allocation requirements.

The trend in the distribution of errors of Fig. 4 depicts the association between the forecasted values and residual values that can be utilized to gauge the bias in the model. The clustering of the data around the horizontal axis at zero is heavy, and the majority of the errors lie within a narrow range between −0.1 and 0.1. Lack of apparent trends in the dispersion of data points indicates that the model is homoscedastic, and it has done excellently in capturing the underlying data trends. The reliability of the model can be confirmed by the consistent performance within the range of 0.05 to 0.65.

The performance benchmarks shown in Fig. 5 provide a clear evaluation of the model's predictive accuracy using standard regression metrics. The obtained values for MAE, MSE, and RMSE are 0.0306, 0.0018, and 0.0429, respectively, indicating strong model performance. These low error values suggest that the predicted outputs closely match the actual throughput values, reflecting high precision in forecasting. Furthermore, the minimal MSE highlights the model's stability, showing that it is not significantly affected by large prediction deviations. Overall, these

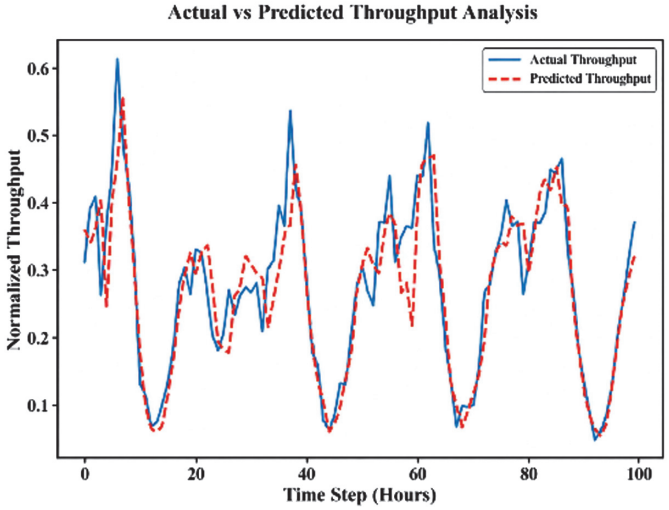


Fig. 3. Comparison of actual vs. predicted throughput.



Fig. 4. Residual error analysis of throughput predictions.

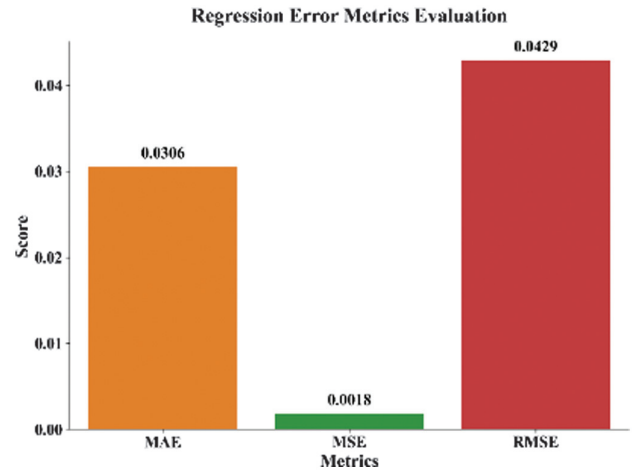


Fig. 5. Quantitative regression error metrics evaluation.

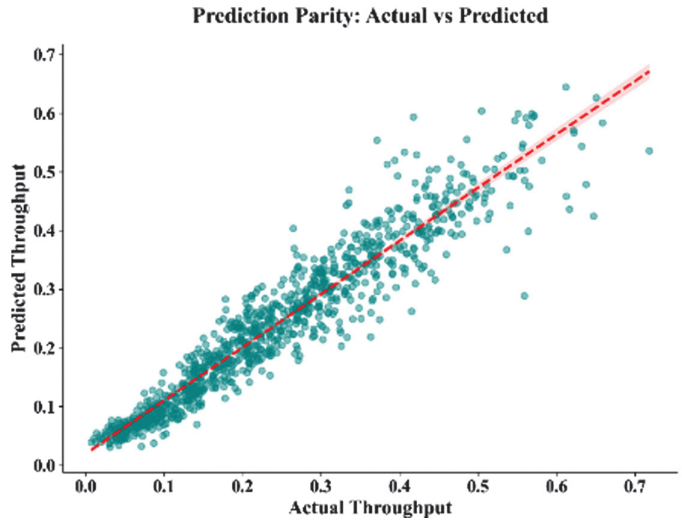


Fig. 6. Prediction parity and regression correlation analysis.

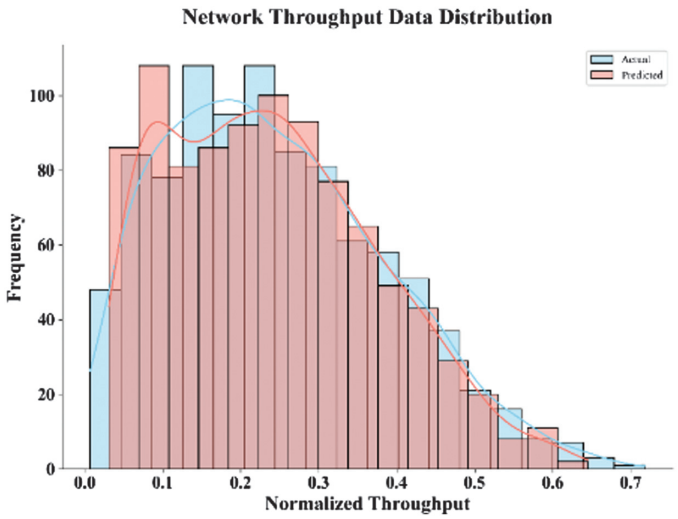


Fig. 7. Comparative distribution of actual vs. predicted throughput.

results demonstrate that the model is both accurate and reliable, making it well suited for effective throughput prediction in dynamic network environments.

Figure 6 illustrates a scatter plot comparing actual and predicted throughput values. Most data points are tightly clustered along the line of identity, particularly within the range of 0.0 to 0.7, indicating strong agreement between observed and predicted values. This close alignment reflects the model's ability to capture underlying patterns in the data with high accuracy. The evident linear relationship suggests a high R^2 value, confirming that the model maintains consistent predictive performance. Importantly, this accuracy is sustained across both low and high traffic conditions, demonstrating the model's robustness and reliability in handling varying network intensities without significant deviation.

Figure 7 illustrates a comparison between actual and predicted data using two visualization methods: a histogram and a Kernel Density Estimate (KDE). Both distributions exhibit a pronounced rightward skew, with the majority of normalized throughput values falling between 0.1 and 0.3. This indicates that most network throughput measurements are concentrated in this lower range. The close alignment and significant overlap between the light blue (actual) and salmon (predicted) areas suggest that the predictive model accurately captures the underlying statistical characteristics of the dataset. Moreover, it demonstrates that the model preserves the inherent variability in the data, reflecting its reliability in representing both central tendencies and distributional nuances of the original measurements.

The learning curves in Fig. 8 illustrate the model's performance across 65 training epochs, highlighting both convergence and generalization. The training loss, represented by the blue line, steadily declines from an initial value of 0.0095 to approximately 0.0022, reflecting effective learning. Meanwhile, the validation loss, shown as the orange line, stabilizes around 0.0038, indicating that the model maintains consistent performance on unseen data. This trend demonstrates that the system successfully mitigates overfitting while preserving predictive accuracy. Overall, the curves confirm that the model not only learns efficiently from the training data but also generalizes well, ensuring reliable predictions across diverse input scenarios.

In Fig. 9, the relationship between normalized packet count and normalized byte length is clearly evident. The plot demonstrates a

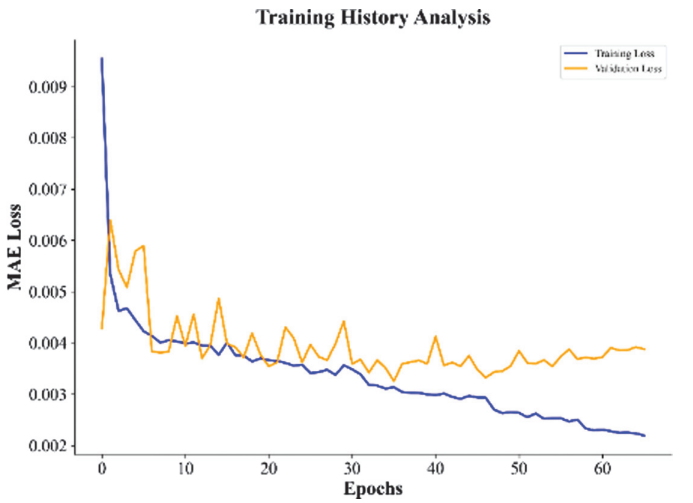


Fig. 8. Convergence analysis of model training and validation loss.

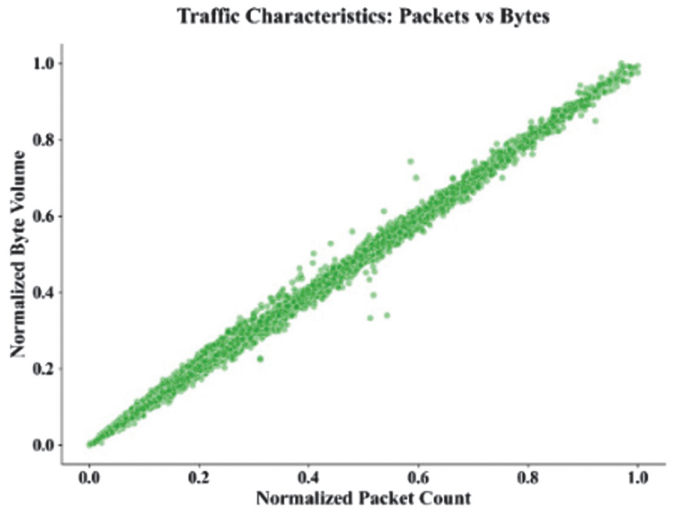


Fig. 9. Correlation analysis of network traffic characteristics.

nearly perfect correlation across the 0.0 to 1.0 range, indicating that packet sizes remain consistent and stable regardless of network density. This consistency suggests that either metric packet count or byte length can reliably be used for throughput prediction. The strong correlation not only confirms the predictability of network traffic but also reflects the high quality of the dataset. Overall, these observations validate that the dataset provides a robust foundation for modeling and forecasting network performance, supporting accurate and dependable throughput analysis.

C. EVALUATION OF COMMUNICATION PERFORMANCE UNDER DIFFERENT LOAD CONDITIONS

The results of the system-level evaluation under low, medium, and high traffic loads are presented in Table II. The baseline and proposed BiLSTM-assisted approaches are compared in terms of average throughput, latency, packet loss, and resource utilization.

As shown in Table II, the proposed approach improves performance across all three load conditions. Throughput increases from 0.52 to 0.56 under low load, from 0.43 to 0.48 under medium load, and from 0.31 to 0.38 under high load. Latency and packet loss also decrease across all conditions, with maximum improvements of 22.58% in throughput, 23.29% in latency reduction, and 37.93% in packet-loss reduction under high-load conditions.

D. COMPARATIVE ANALYSIS

This section is intended for a comparative analysis of the effectiveness of the developed model BiLSTM network throughput prediction framework with respect to recent traffic prediction approaches. This comparison of the evaluation criteria, results, and application areas of the respective studies is intended for a better understanding of the approaches for effective network traffic prediction and their application for improved communication.

Table III compares the proposed BiLSTM-based framework with the existing traffic prediction approaches, highlighting the efficiency of the proposed framework. In fact, the existing approaches have focused on distributed learning, privacy preservation, and large-scale network optimization. These approaches often lead to the incorporation of extra computational complexities. However, the proposed model learns the temporal dependencies of the network traffic data through the direct processing of the bidirectional sequence. This results in accurate and stable predictions of the network throughput. This is confirmed through the low error rates

Table II. System-level communication-efficiency evaluation under different network load conditions

Load condition	Method	Average throughput (normalized)	Average latency (ms)	Packet loss (%)	Resource utilization (%)	Throughput improvement (%)	Latency reduction (%)	Packet loss reduction (%)
Low	Baseline	0.52	18.4	2.6	72.3	–	–	–
Low	Proposed BiLSTM-assisted approach	0.56	15.1	1.8	80.1	7.69	17.93	30.77
Medium	Baseline	0.43	31.7	4.8	68.5	–	–	–
Medium	Proposed BiLSTM-assisted approach	0.48	25.4	3.1	78.6	11.63	19.87	35.42
High	Baseline	0.31	49.8	8.7	61.2	–	–	–
High	Proposed BiLSTM-assisted approach	0.38	38.2	5.4	74.5	22.58	23.29	37.93

Table III. Comparative analysis of network traffic prediction approaches

Method	Evaluation metrics	Key result	Application focus	Performance outcome
Proposed BiLSTM-based throughput prediction model	MAE, MSE, RMSE	Accurate throughput prediction	Network communication efficiency	Reliable traffic forecasting
ST-GNN with federated learning [31]	MAE, RMSE	Improved spatio-temporal dependency learning	Traffic flow prediction	Enhanced pattern modeling
Split learning + federated learning [32]	Prediction accuracy, latency	Improved distributed prediction	Wireless traffic prediction	Privacy-preserving forecasting
Federated learning traffic prediction framework [33]	Prediction accuracy, energy consumption	Improved VNF decision support	Edge–cloud network management	Optimized resource utilization

Table IV. Quantitative performance comparison of different models under the same experimental conditions

Model	Dataset	Task	Sliding window	MAE	MSE	RMSE	R^2	MAE improvement vs. LSTM (%)	RMSE improvement vs. LSTM (%)
LSTM	CESNET-TimeSeries24	Network throughput prediction	10	0.0398	0.0022	0.0469	0.9635	0	0
GRU	CESNET-TimeSeries24	Network throughput prediction	10	0.0371	0.0020	0.0447	0.9668	6.78	4.69
SVR	CESNET-TimeSeries24	Network throughput prediction	10	0.0452	0.0029	0.0539	0.9517	−13.57	−14.93
Random Forest	CESNET-TimeSeries24	Network throughput prediction	10	0.0415	0.0025	0.0500	0.9584	−4.27	−6.61
Proposed BiLSTM	CESNET-TimeSeries24	Network throughput prediction	10	0.0306	0.0018	0.0429	0.9716	23.12	8.53

(MAE: 0.0306, MSE: 0.0018, and RMSE: 0.0429). The proposed framework is reliable and efficient network-traffic prediction with increased accuracy.

A quantitative comparison was conducted under consistent experimental conditions by evaluating the proposed BiLSTM model against LSTM, GRU, support vector regression (SVR), and Random Forest models using the CESNET-TimeSeries24 dataset. All models were configured using the same sliding-window size and evaluated using MAE, MSE, RMSE, and R^2 to provide a consistent basis for comparison.

As shown in Table IV, the proposed BiLSTM achieved an MAE of 0.0306, MSE of 0.0018, RMSE of 0.0429, and R^2 of 0.9716. Compared with LSTM, the proposed model achieved 23.12% lower MAE and 8.53% lower RMSE, while obtaining the highest R^2 among the compared models.

E. DISCUSSION

The findings indicate that the proposed BiLSTM-based framework effectively captures the temporal dependencies inherent in network communications, enabling accurate throughput predictions across a range of scenarios. By leveraging a bidirectional learning approach, the model can identify complex sequential patterns that might be missed by unidirectional or simpler models, resulting in more consistent and reliable performance. This underscores the importance of temporal DL models in modern network management, as they allow for data-driven insights that can optimize communication efficiency and resource allocation. Compared to alternative methods that rely on distributed learning or federated learning techniques, which often involve higher computational complexity and coordination overhead, the BiLSTM approach offers a simpler yet robust solution. Its ability to process sequences in both forward and backward directions ensures that past and future context is utilized, enhancing prediction accuracy without adding unnecessary complexity. The results demonstrate the applicability of machine learning for network intelligence through quantitative evaluation of network throughput prediction under sequential traffic conditions. The primary contribution of this work lies in the practical application of an established BiLSTM architecture together with a structured data-processing pipeline for network throughput forecasting using the CESNET-TimeSeries24 dataset. The study further provides a quantitative evaluation of the framework using standard regression metrics. Thus, the contribution is primarily application- and evaluation-oriented rather than a methodological innovation of the BiLSTM architecture.

The CESNET-TimeSeries24 dataset provides large-scale time-series network traffic statistics and is suitable for evaluating network throughput prediction. However, it does not directly contain system-level information related to resource allocation, network optimization, QoS, latency, congestion, or packet loss. Consequently, the regression results establish the predictive capability of the proposed BiLSTM framework, while the separate system-level evaluation in Section IV.C provides additional evidence regarding throughput, latency, packet loss, and resource utilization under different network load conditions. However, broader aspects such as resource allocation, QoS, and network optimization require further validation. The potential use of throughput predictions for communication-efficiency-oriented network management therefore requires additional system-level validation.

V. CONCLUSION

This study proposes a framework based on DL techniques for predicting network throughput using the CESNET-TimeSeries24 dataset, with a particular focus on capturing complex sequential patterns in network communication data. A BiLSTM model was employed to leverage bidirectional learning, enabling the system to better understand both past and future dependencies in the data compared to traditional models. A structured preprocessing pipeline was developed to ensure input data quality and consistency, which is crucial for robust model training. Experimental results demonstrate that the proposed model achieves high predictive performance, with error metrics indicating MAE = 0.0306, MSE = 0.0018, and RMSE = 0.0429, highlighting its ability to accurately capture network behavior. Beyond its predictive capabilities, the study emphasizes the balance between model accuracy and implementation simplicity, ensuring that the framework remains practical for network throughput prediction. The key contribution lies in providing a reliable and efficient approach for network throughput prediction based on the CESNET-TimeSeries24 dataset. However, the dataset does not provide direct system-level information for evaluating resource allocation, network optimization, latency, congestion, packet loss, or QoS. Therefore, the present study validates the predictive capability of the BiLSTM framework and provides supplementary system-level evaluation of throughput, latency, packet loss, and resource utilization under different network load conditions. Broader communication-efficiency aspects, including resource allocation, QoS, and network optimization, require further validation. The predicted throughput information may potentially support traffic management and communication-efficiency-oriented network operations, but these downstream benefits require dedicated system-level validation.

A. FUTURE WORK

Future research may extend this work by exploring hybrid DL architectures that combine BiLSTM with attention and transformer-based models to improve prediction accuracy. The framework needs to be tested on bigger network traffic datasets which have different types of data to assess its scalability and ability to generalize. Future work will extend the evaluation beyond the current throughput and system-level analyses by incorporating additional network datasets and more comprehensive network management experiments. In particular, further studies will investigate resource allocation, QoS, congestion behavior, and network optimization across diverse and realistic network conditions.

DATA AVAILABILITY STATEMENT

The data is available upon reasonable request.

FUNDING STATEMENT

The authors received no specific funding or financial support from public, commercial, or non-profit organizations for the research, authorship, or publication of this article.

AUTHOR CONTRIBUTION

YuSheng Jiang: Conceptualization, methodology, data preprocessing, model development, experimentation, analysis, and manuscript preparation. Muhan Li: Data analysis, validation, interpretation of results, and manuscript review. Ni Liang: Methodological review, supervision, manuscript editing, and critical revision. All authors contributed to the final manuscript and approved the submitted version.

ETHICAL APPROVAL

Ethical approval was not required for this study because it exclusively used publicly available, anonymized network traffic data and did not involve human participants, human subjects, or personally identifiable information.

CONSENT TO PARTICIPATE

Not applicable, as this study did not involve human participants.

CONSENT TO PUBLICATION

Not applicable, as the study does not contain identifiable information about individual participants.

COMPETING INTERESTS

The authors declare that they have no competing interests related to this work.

ACKNOWLEDGMENTS

The authors would like to thank the providers and maintainers of the CESNET-TimeSeries24 dataset for making the network traffic data publicly available. The authors also acknowledge the support of their institution during the completion of this research.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] G. A. Senthil, R. Prabha, and R. Renuka Devi, "Energy efficient multipath routing in IoT-wireless sensor network via hybrid optimization and deep learning-based energy prediction," *Netw.: Comput. Neural Syst.*, vol. 0, no. 0, pp. 1–50, Apr. 2025, DOI. 10.1080/0954898X.2025.2476081.

[2] F. B. Teixeira *et al.*, "Converge: towards an efficient multi-modal sensing research infrastructure for next-generation 6 G networks," *J. Wirel. Commun. Netw.*, vol. 2026, no. 1, p. 18, Dec. 2025, DOI. 10.1186/s13638-025-02544-y.

[3] K. Saleem *et al.*, "Intelligent multi-agent model for energy-efficient communication in wireless sensor networks," *EURASIP J. Info. Secur.*, vol. 2024, no. 1, p. 9, Apr. 2024, DOI. 10.1186/s13635-024-00155-6.

[4] M. N. Fauzan *et al.*, "Enhanced wireless sensor network lifetime using EGWO-optimized neural network approach," *Network*, vol. 6, no. 1, p. 5, Mar. 2026, DOI. 10.3390/network6010005.

[5] W. Symbor and Ł. Falas, "Ensuring reliable network communication and data processing in Internet of Things Systems with prediction-based resource allocation," *Sensors*, vol. 25, no. 1, p. 247, Jan. 2025, DOI. 10.3390/s25010247.

[6] A. Mystakidis, P. Koukaras, and C. Tjortjis, "Advances in traffic congestion prediction: An overview of emerging techniques and methods," *Smart Cities*, vol. 8, no. 1, Feb. 2025, DOI: 10.3390/smartcities8010025.

[7] H. Yu *et al.*, "Renyi entropy-driven network traffic anomaly detection with dynamic threshold," *Cybersecurity*, vol. 7, no. 1, p. 64, Dec. 2024, DOI. 10.1186/s42400-024-00249-1.

[8] R. Abduljabbar, H. Dia, and S. Liyanage, "Machine learning models for traffic prediction on arterial roads using traffic features and weather information," *Appl. Sci.*, vol. 14, no. 23, p. 11047, Jan. 2024, DOI. 10.3390/app142311047.

[9] A.-L. Toba *et al.*, "Long-term traffic prediction using deep learning long short-term memory," *Smart Cities*, vol. 8, no. 4, Jul. 2025, DOI. 10.3390/smartcities8040126.

[10] M. L. Francies *et al.*, "A framework for continual learning in real-time traffic forecasting utilizing spatial–temporal graph convolutional recurrent networks," *Complex Intell. Syst.*, vol. 11, no. 10, p. 420, Aug. 2025, DOI. 10.1007/s40747-025-02049-7.

[11] R. K. Udumu and D. Vasumathi, "Software defect prediction using temporal transformer graph network with newton–raphson optimization," *J. Artif. Intell. Tech.*, vol. 5, pp. 441–452, Oct. 2025, DOI. 10.37965/jait.2025.0751.

[12] J. Kim and I. Ryoo, "A deep dive into ai-based network traffic prediction using heterogeneous real datasets," *Appl. Sci.*, vol. 16, no. 1, p. 367, Jan. 2026, DOI. 10.3390/app16010367.

[13] L. L. Prasanth and E. Uma, "A computationally intelligent framework for traffic engineering and congestion management in software-defined network (SDN)," *J. Wirel. Commun. Netw.*, vol. 2024, no. 1, p. 63, Aug. 2024, DOI. 10.1186/s13638-024-02392-2.

[14] M. Magdy Saady *et al.*, "Deep learning neural networks-based traffic predictors for V2X communication networks," Front. *Artif. Intell.*, vol. 8, Dec. 2025, DOI. 10.3389/frai.2025.1701951.

[15] Q. Cui *et al.*, "Overview of AI and communication for 6G network: fundamentals, challenges, and future research opportunities," *Sci. China Inf. Sci.*, vol. 68, no. 7, p. 171301, Apr. 2025, DOI. 10.1007/s11432-024-4337-1.

[16] M. El-Hajj, "Enhancing communication networks in the new era with artificial intelligence: Techniques, applications, and future directions," *Network*, vol. 5, no. 1, p. 1, Mar. 2025, DOI. 10.3390/network5010001.

[17] Y. E. Tok *et al.*, "Artificial intelligence for next-generation 6G technologies and networks," *Discov. Netw.*, vol. 2, no. 1, p. 3, Feb. 2026, DOI. 10.1007/s44354-026-00016-3.

[18] C. Venkatesan, "AI-enhanced routing and slicing strategy for QoS-aware mobile ad hoc networks," *Sci. Rep.*, vol. 16, no. 1, p. 4468, Jan. 2026, DOI. 10.1038/s41598-025-34588-x.

[19] S. B. Khan *et al.*, "Artificial intelligence in next-generation networking: Energy efficiency optimization in IoT networks using hybrid LEACH protocol," *SN. Comput. Sci.*, vol. 5, no. 5, p. 546, May 2024, DOI. 10.1007/s42979-024-02778-5.

[20] W. A. Aziz *et al.*, "DSTL: A dual-step transfer learning-based prediction model for next-generation intelligent cellular networks," *Intell. Converg. Netw.*, vol. 6, no. 1, pp. 82–101, Mar. 2025, DOI. 10.23919/ICN.2025.0005.

[21] Z. Zakariya Saleh, M. F. Abbod, and R. Nilavalan, "Intelligent resource allocation via hybrid reinforcement learning in 5G network slicing," *IEEE Access*, vol. 13, pp. 47440–47458, 2025, DOI. 10.1109/ACCESS.2025.3550518.

[22] A. Aleran *et al.*, "Artificial intelligence and the Internet of Things in energy preservation: Research prototypes, trends, and future directions," *Computing*, vol. 107, no. 11, p. 209, Oct. 2025, DOI: 10.1007/s00607-025-01567-z.

[23] M. M. Kamal *et al.*, "Hybrid optimization-based deep learning for energy efficiency resource allocation in MIMO-enabled wireless networks," *Sci. Rep.*, vol. 15, no. 1, p. 31642, Aug. 2025, DOI. 10.1038/s41598-025-16571-8.

[24] B. Saoud *et al.*, "Mobility and handover management in 5G/6G networks: Challenges, innovations, and sustainable solutions," *Technologies*, vol. 13, no. 8, p. 352, Aug. 2025, DOI. 10.3390/technologies13080352.

[25] S. Konatam, N. Dulam, and P. Raghavan, "AI-Based traffic prediction and load balancing in wireless networks," *Int. J. Glob. Innov. Solut. (IJGIS)*, Nov. 2024, pp. 1-2, DOI. 10.21428/e90189c8.719664c4.

[26] W. Jiang *et al.*, "Graph neural networks for routing optimization: Challenges and opportunities," *Sustainability*, vol. 16, no. 21, p. 9239, Jan. 2024, DOI. 10.3390/su16219239.

[27] X. Duan *et al.*, "AI-agent communication network for 6G: vision, architecture, and key technologies," *Front. Inform. Technol. Electron. Eng.*, vol. 26, no. 11, pp. 2065–2080, Nov. 2025, DOI. 10.1631/FITEE.2500582.

[28] N. Omheni, H. Koubaa, and F. Zarai, "Artificial intelligence for 5G and 6G Networks: A taxonomy-based survey of applications, trends, and challenges," *Technologies*, vol. 13, no. 12, p. 559, Dec. 2025, DOI. 10.3390/technologies13120559.

[29] F. Banaie *et al.*, "Energy efficiency support for software defined networks: a serverless computing approach," *Future Gener. Comput. Syst.*, vol. 176, p. 108121, Mar. 2026, DOI. 10.1016/j.future.2025.108121.

[30] J. Koumar *et al.*, "CESNET-TimeSeries24: Time series dataset for network traffic anomaly detection and forecasting," Zenodo, Sep. 2024, DOI. 10.5281/zenodo.13382427.

[31] J. Feng, C. Du, and Q. Mu, "Traffic flow prediction based on federated learning and spatio-temporal graph neural networks," *ISPRS Int. J. Geo-Inf.*, vol. 13, no. 6, p. 210, Jun. 2024, DOI. 10.3390/ijgi13060210.

[32] P. Li *et al.*, "Core network traffic prediction based on vertical federated learning and split learning," *Sci. Rep.*, vol. 14, no. 1, p. 4663, Feb. 2024, DOI. 10.1038/s41598-024-53193-y.

[33] Y. Hu *et al.*, "Network traffic prediction in an edge–cloud continuum network for multiple network service providers," *Electronics*, vol. 13, no. 17, p. 3515, Jan. 2024, DOI. 10.3390/electronics13173515.