International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 627-635
https://doi.org/10.70917/ijcisim-2026-5070
Article

Modelling and predicting Computer Network using ARIMA Models and Neural Network Method

Fatima Yahya Mukhtar[1], Adil Eltayab Yousif[2]

[1]Faculty of Math Sc and Informatics, Khartoum University, Khartoum, Sudan
[2]Faculty of Math Sc and Informatics, Khartoum University, Khartoum, Sudan
[1]st fatimayahia423@gmail.com; 2ndadilyousif202@gmail.com;
[1]https://orcid.org/0009-0001-8153-712X ; 2 https://orcid.org/0009000912055117;

Abstract: This study compares ARIMA-based statistical models and RNN-based neural models for forecasting hourly network traffic. Four datasets—simulated, daily-Dominant, Weekly-Dominant, and Balanced-Dominant—are analyzed using spectral diagnostics to identify daily and weekly periodicities. Three MSARIMA variants, Auto-ARIMA, a tanh-RNN, and a mean-activation RNN are evaluated using MAE, RMSE, MAPE, and R^2. MSARIMA models consistently achieve the highest accuracy, with R^2 above 0.97 in simpler datasets. RNN models severely underfit, producing "extremely high error rates exceeding 90% MAPE. The results show that MSARIMA is a reliable solution for short-term traffic forecasting, while simple RNNs fail to capture seasonal structure.
Keywords: ARIMA, Network Traffic, RNN, Spectral Analysis, Time Series

1. Introduction

Computer networks have become indispensable to modern organizations and governmental institutions, to the point that daily operations cannot be sustained without reliable networking infrastructures [1]. Over the past decade, the number of devices directly connected to the Internet has grown at an unprecedented rate. As a global "network of networks," the Internet interconnects billions of hosts and users worldwide, enabling the transmission of massive volumes of data across its backbone.

Forecasts by the International Telecommunication Union (ITU) estimated that nearly three billion users would be connected to the Internet by early 2015. This rapid expansion has heightened the importance of Internet traffic prediction, drawing increasing attention from the telecommunications and computer networking research communities. Accurate traffic forecasting directly contributes to improved Quality of Service (QoS) by supporting effective network design, management, planning, control, and optimization. Prior studies have shown that both linear and nonlinear time series models are widely used for predicting network traffic behaviour [2].

Internet traffic measurement and analysis produce datasets that reflect usage trends and network behaviour. These datasets serve as essential inputs for forecasting models based on statistical and analytical techniques [3]. Despite its importance, Internet traffic forecasting has received relatively limited attention within the computer networks domain. Enhancing prediction accuracy enables the development of more efficient traffic engineering and anomaly detection tools, ultimately yielding economic benefits through improved resource utilization and management. Over the past few decades, numerous network traffic models have been proposed to capture two fundamental characteristics commonly observed in traffic data: statistical properties and self-similarity.

While these models have achieved varying degrees of success in representing variability and long-range dependence, comparatively little attention has been given to periodic characteristics—such as daily and weekly cycles—despite their strong presence in real operational networks [4].

In recent years, Artificial Neural Networks (ANNs) have gained significant popularity as effective tools for classification, clustering, pattern recognition, and prediction across scientific and engineering domains. ANN performance is typically evaluated in terms of prediction accuracy, processing speed, latency, scalability, fault tolerance, and convergence behaviour. Owing to their self-learning capability, adaptability, nonlinear mapping ability, and robustness to noise, ANNs are widely used for universal function approximation in numerical and time-series prediction tasks.

Neural networks can be broadly categorized into feed-forward neural networks (FFNNs) and feedback (recurrent) neural networks. FFNNs consist of layers of interconnected processing units through which information flows unidirectionally from input to output. Each connection carries a weight representing the learned relationship between neurons, and Multi-Layer Perceptron (MLP) networks are among the most widely used FFNN architectures in traffic prediction studies [5].

Feedback or recurrent neural networks incorporate internal state or memory, enabling them to process sequential data and capture temporal dependencies. These networks are particularly suitable for time-series forecasting, pattern recognition, and sequence modelling. Recurrent connections create cycles within the network, allowing it to exhibit dynamic temporal behaviour. A prominent example is the Recurrent Neural Network (RNN), which processes sequences by feeding information from previous time steps back into the model [6]. Advanced variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were developed to address limitations like vanishing gradients and to improve the modelling of long-range dependencies.

Compared with traditional statistical approaches, ANNs demonstrate strong capability in modelling nonlinear time series and handling noisy data. Although various neural architectures have been applied to traffic forecasting, early studies primarily focused on MLP models, with more recent work emphasizing recurrent and deep learning architectures.

Cortez, Rio, Rocha, and Sousa proposed a Neural Network Ensemble (NNE) approach for predicting TCP/IP traffic using a time-series forecasting framework. Their experiments, conducted on real traffic data from two large Internet Service Providers and evaluated at five-minute and hourly intervals, showed that the NNE approach achieved competitive performance relative to traditional forecasting methods such as Holt–Winters and ARIMA. However, the study assumed passive monitoring and did not address active measurement scenarios involving real-time packet-level data [7].

Belhaj and Tagina investigated Internet end-to-end delay prediction using Recurrent Neural Networks, employing Round-Trip Time (RTT) as the primary forecasting metric. Traffic delay data were collected over several days and divided into training and testing subsets. Their RNN-based model demonstrated strong adaptability and accuracy in tracking RTT dynamics, including multi-step-ahead predictions, highlighting the effectiveness of RNNs in modelling nonlinear network delay behaviour [8].

2. Methodology

This study evaluates two modelling paradigms for network traffic forecasting: the Autoregressive Integrated Moving Average (ARIMA) family of models and Recurrent Neural Networks (RNNs). ARIMA models are well established in time-series analysis and excel at capturing linear dependencies, trends, and seasonal patterns under stationarity assumptions. In contrast, RNN-based models can learn nonlinear temporal dynamics and long-range dependencies through their internal memory mechanisms. A unified experimental framework is introduced to ensure a fair and consistent comparison between ARIMA and RNN models. The framework encompasses data pre-processing, stationarity assessment, frequency-domain analysis, model construction, and validation across multiple datasets. Model Performance is evaluated using standard accuracy metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The framework begins with the acquisition of network traffic data from a real network environment consisting of homogeneous nodes operating under stable conditions. The collected measurements represent packet-volume time series sampled at uniform intervals, making them suitable for both time-domain and frequency-domain analysis. Following acquisition, the data undergo cleaning, formatting, visualization, and division into training and testing subsets. The training set is used to construct and calibrate forecasting models, while the testing set is reserved exclusively for validation.

Pre-processing includes examining the statistical properties of the traffic series to assess stationarity. Objective tests such as the Augmented Dickey–Fuller (ADF) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests determine whether differencing is required. When non-stationarity is detected, appropriate differencing operations are applied. Autocorrelation and partial autocorrelation structures are further explored using ACF and PACF plots to reveal underlying temporal dependencies.

A distinctive component of the framework is the incorporation of spectral analysis prior to model construction. By computing the periodogram and analysing the detrended spectrum, dominant frequency components—particularly daily and weekly cycles—are identified. These frequency-domain insights guide the selection of seasonal parameters for ARIMA-based models and help interpret the temporal patterns learned by neural network models. Peaks in the periodogram indicate strong periodic components that significantly contribute to traffic variation.

Model performance is then assessed using multiple evaluation metrics (MAE, RMSE, MAPE, and R^2) to ensure a comprehensive comparison of accuracy and robustness. The framework concludes with a comparative analysis stage, where the forecasting performance of ARIMA and RNN models is systematically evaluated.

A. Autoregressive Integrated Moving Average (ARIMA) Models

This section presents the statistical modelling approach adopted in this research using the Autoregressive Integrated Moving Average (ARIMA) family of models. ARIMA models are among the most widely used techniques for time series forecasting due to their solid theoretical foundation and effectiveness in modelling linear temporal dependencies, trends, and seasonal patterns under stationarity assumptions.

An ARIMA model is characterized by three parameters, (p, d, q), where p denotes the order of the autoregressive (AR) component, d represents the degree of differencing required to achieve stationarity, and q corresponds to the order of the moving-average (MA) component. The general form of an ARIMA (p, d, q) model can be expressed as:

$$\phi(L)(1-L)^d Y^t = \theta\,(L)E_t$$

where Y^t denotes the observed network traffic at time t, L is the lag operator, $\phi(L)$ and $\theta\,(L)$ are polynomials of orders p and q, respectively, and $E_t$is a white noise error term which normally distributed with zero mean and constant variance.

In the autoregressive component, the current value of the time series is modelled as a linear combination of its previous values. An AR process of order p, denoted as AR(p), is defined as:

$$X_t = \varphi_1 X_{t-1} + \cdots + \varphi_p X_{t-p} + Z_t$$

where φ^i's are model parameters and Z_t is a white noise process.

Similarly, the moving-average component models the current value as a linear combination of past error terms. A moving-average process of order q, denoted as MA(q), is given by:

$$X_t = Z_t + \beta_1 Z_{t-1} + \cdots + \beta_{t-q} Z_{t-q}$$

where $\beta' s$ are constant coefficients.

The combination of AR and MA components results in the ARMA (p, q) model, which serves as the foundation for ARIMA modelling after differencing is applied.

Since ARIMA models require stationary input series, stationarity assessment is a critical step prior to model fitting. Stationarity is evaluated using the Augmented Dickey–Fuller (ADF) and KPSS tests. If the original network traffic series is found to be non-stationary, differencing is applied to remove trends and stabilize the mean.

Let the differenced series be W_t and d be the number of difference transformations then:

$W_t = \nabla^d X_t = (1 - B)^d X_t$ and W_t can be written as:

$$W_t = \varphi_1 W_{t-1} + .. + \varphi_p W_{t-p} + Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}$$

$$Z_t = WN(0, \sigma^2)$$

Model identification involves selecting appropriate values of p and q. In this study, preliminary guidance is obtained from the ACF and PACF plots, which indicate significant lag structures.

To select the most suitable model, the Akaike Information Criterion (AIC) is employed. AIC balances model goodness-of-fit with model complexity by penalizing excessive numbers of parameters. The model with the minimum AIC value is selected as the optimal ARIMA specification.

To detect cycles, periodicities, and structural patterns that may be hidden in the time domain spectral analysis is used to examine how the *variance* of a time series is distributed across different frequencies.

B. Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specifically designed to model sequential and temporal data. Unlike feedforward neural networks, RNNs incorporate feedback connections that allow information from previous time steps to influence the current output. This internal memory mechanism makes RNNs particularly suitable for time series forecasting problems such as network traffic prediction.

In this study, RNN models are employed to capture nonlinear temporal dependencies in network traffic data and to compare their forecasting performance with statistical ARIMA-based models. Two RNN variants are investigated: a standard RNN with a hyperbolic tangent (tanh) activation function and a modified RNN incorporating a mean-based activation function derived from traffic characteristics at different times of the day.

An RNN processes an input sequence by maintaining a hidden state that evolves over time. At each time period t, the hidden state is updated based on the current input and the previous hidden state.

Hidden-state update:

$h_t = f\ (h_{t-1}, x_t)$, where:

h_t: current hidden state at time t

h_{t-1}: previous hidden state

x_t: input at time t

Activation function:

$h_t = g\ (W_{hh} h_{t-1} + W_{xh} x_t)$, where:

g(·) is a smooth, bounded activation function such as the sigmoid or hyperbolic tangent.

W_{hh}denotes the recurrent weight parameter(s).

$W_{xh} x_t$ denotes the input weight parameter(s).

In the first RNN configuration, the hyperbolic tangent (tanh) activation function is used to update the hidden state. The tanh function is widely adopted due to its ability to map inputs into a bounded range [−1,1], which helps stabilize training and mitigate gradient explosion.

The tanh activation function is defined as:

g(z) = tanh(z)

Substituting into the hidden state equation yields:

$$h_t = tanh\,(W_{hh}h_{t-1} + W_{xh}x_t)$$

This standard RNN formulation serves as a baseline deep learning model for capturing nonlinear temporal dynamics in network traffic data.

To further enhance the ability of the RNN to reflect traffic behaviour patterns, this study proposes a modified activation mechanism based on the mean traffic level observed during specific time periods. Instead of relying solely on conventional nonlinear activation functions, the activation output is influenced by the statistical mean of the input traffic corresponding to different time-of-day segments, such as early morning, afternoon, evening, and night.

Let k denote the mean traffic value for time segment k. The mean-based activation function can be represented as:

g (mean)(Z_t) = k

Accordingly, the hidden state update becomes:

h_t= g (mean)($W_{hh}h_{t-1} + W_{xh}x_t$)

This approach introduces domain-specific knowledge into the RNN structure by incorporating temporal traffic characteristics directly into the activation mechanism. The objective is to investigate whether such an activation strategy can improve forecasting performance compared to the standard tanh-based RNN, particularly in datasets with strong periodic patterns.

Both RNN variants are trained using historical network traffic observations. The same training and testing datasets are used to ensure a fair comparison between models. During training, the network parameters are optimized to minimize prediction error, while during testing, the trained RNN models generate forecasts for unseen data.

The forecasting performance of the RNN models is evaluated using the same error metrics applied to ARIMA-based models, enabling a direct and consistent comparison between statistical and deep learning approaches.

3. Analysis

The primary objective of this study is to evaluate the performance of statistical and deep learning–based forecasting models under varying levels of data complexity and temporal characteristics. The analysis is conducted on four datasets: a simulated baseline dataset and three real-world datasets exhibiting Daily-Dominant, Weekly-Dominant, and balanced seasonal behaviours. These datasets represent progressively increasing levels of temporal complexity compared to the initial simulated data.

A. Spectral Analysis of the Balanced-Dominant Dataset

Based on the spectral analysis results, the network traffic data exhibit strong cyclic patterns, particularly daily and weekly periodicities. To account for these effects, Seasonal ARIMA models, denoted as ARIMA × (p,d,q) s ×(P,D,Q) are employed.

In this formulation, (P, D, Q) are the seasonal autoregressive, differencing, and moving-average orders, respectively, and s denotes the seasonal period. In this research, seasonal periods are selected according to the dominant frequencies identified through spectral analysis, such as s = 24 for daily cycles and s = 168 for weekly cycles in hourly traffic data.

The Balanced-Dominant dataset exhibits the most complex spectral structure among the four datasets.

After detrending, the periodogram reveals a clearer separation between the daily and weekly frequency components. Although both periodicities remain present, neither dominates the spectrum entirely. This multi-seasonal behavior confirms the complex nature of the balanced dataset and highlights the challenge of accurately modeling such data.

The spectral complexity observed in this dataset explains the increased difficulty faced by both ARIMA-based and RNN-based models. While seasonal ARIMA models benefit from explicitly defined seasonal parameters, capturing multiple interacting cycles remains a challenging task, particularly for neural networks trained without explicit frequency-domain guidance.

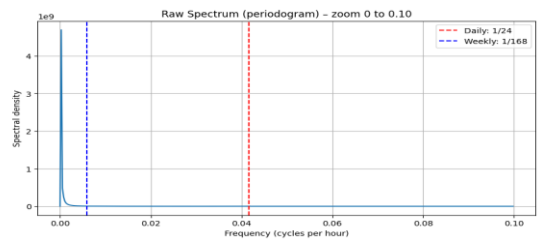

Fig [2]: Detrended Spectrum.

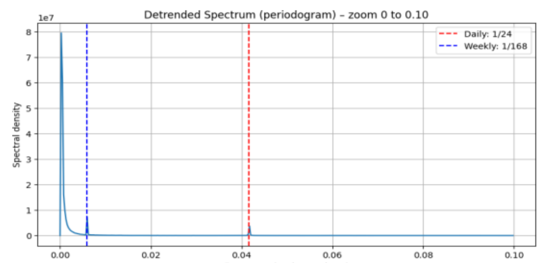

Fig [2]: Detrended Spectrum.

Across all four datasets, spectral analysis provides critical insights into the underlying temporal dynamics that govern network traffic behavior. The results demonstrate that datasets with clearly dominant and stable frequency components—such as the daily- and Weekly-Dominant datasets—are particularly well-suited for ARIMA-based modeling.

Moreover, the detrended spectral analysis plays a crucial role in distinguishing genuine periodic components from trend-induced low-frequency effects. By grounding model selection in frequency-domain evidence, the proposed framework enhances interpretability and explains the observed differences in forecasting performance across datasets

Each dataset was divided using the same training and testing strategy applied in the simulated case, where 80% of the observations were used for training and the remaining 20% were reserved for testing. This consistent splitting strategy ensures comparability across datasets and models.

The following forecasting models were applied to all four datasets:

Small MSARIMA: A seasonal ARIMA model with non-seasonal order (1,0,1) and seasonal order (1,1,1,24). This model represents a relatively simple seasonal structure designed to capture dominant daily periodicity.

Little MSARIMA: A slightly more complex seasonal ARIMA model with order (2,0,2) and the same seasonal configuration (1,1,1,24). This model allows for greater flexibility in capturing short-term dependencies while maintaining the same seasonal behavior.

Auto-ARIMA: An automatically configured ARIMA model in which both non-seasonal and seasonal orders are selected using the Auto-Arima procedure, with the seasonal period set to be 24. The model selection is guided by information criteria and diagnostic checks.

RNN (tanh activation): A recurrent neural network that uses the previous 24-hourly observations as input features to forecast the next step. The tanh activation function is employed to introduce nonlinearity while maintaining bounded outputs.

RNN (mean-based activation): This model follows the same RNN architecture but replaces the tanh activation with the proposed mean-based activation function. The objective is to assess whether incorporating domain-specific temporal aggregation improves forecasting accuracy in datasets with strong seasonal patterns.

Since the four data sets yield to similar results and same conclusion the output of the set of Balanced-Dominant will be displayed in this paper. The results of this set align strongly with the **Daily-Dominant** and **Weekly-Dominant** datasets. In all cases, the **Small MSARIMA** and **Little MSARIMA** models achieve the best performance, maintaining very low errors and demonstrating reliable modeling of trend and multi-seasonality. Meanwhile, **ARIMA/Auto-ARIMA** consistently performs poorly across all datasets, reflecting its inability to capture complex seasonal structures without explicit seasonal components. The **RNN models** (tanh and mean-activation) fail uniformly in the four datasets, producing nearly constant predictions and extremely high errors, confirming that simple RNN architectures cannot learn long-range dependencies or strong seasonal dynamics in traffic data. (Fig [3] – Fig [6])

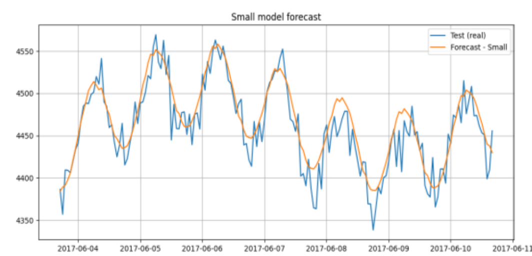

Fig [3]: Small Model Forecast.

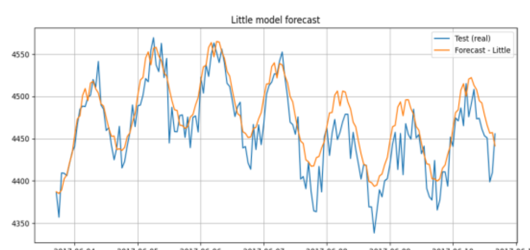

Fig [4]: Little Model Forecast.

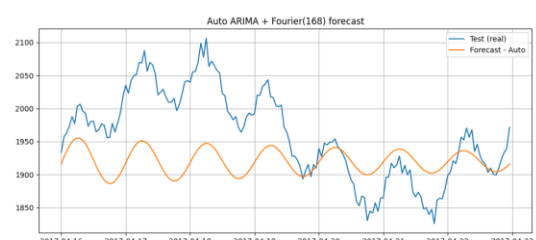

Fig [5]: Auto ARIMA +Fourier (168 hours) Forecast.

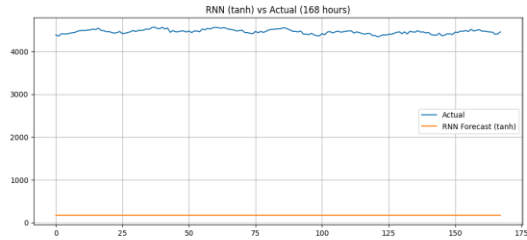

Fig [6]: RNN (tanh) vs Actual (168 hours) Forecast.

TABLE I

Comparative Forecasting Performance Models.

Model	MAE	RMSE	MAPE (%)	R^2	Overall Accuracy (%)
Small (MSARIMA)	16.573	20.947	0.373	0.824	99.63
Little (MSARIMA)	21.062	26.179	0.474	0.725	99.53
Auto-ARIMA	267.013	332.428	6.014	NA	93.99
RNN (tanh)	4284.105	4284.396	96.062	NA	3.94
RNN (mean activation)	4363.495	4363.782	97.842	NA	2.16

Table [1]: Comparative Forecasting Performance Models.

4. Conclusions

ARIMA and seasonal ARIMA models consistently outperformed RNN-based models across all network traffic datasets, even as temporal complexity and seasonality increased. These classical models achieved low forecasting errors and strong goodness-of-fit, demonstrating their ability to capture stable, repetitive temporal patterns. In contrast, RNN models—both standard and mean-activation variants—showed high error rates and unstable performance, particularly in datasets with strong seasonality. The study did not use the Ljung–Box test due to its oversensitivity in large samples, instead relying on residual behavior and forecasting metrics for validation. Overall, ARIMA-based approaches proved more reliable, interpretable, and computationally efficient, while deep learning methods require further refinement before they can match this consistency in structured network traffic forecasting.

REFERENCES

1. *Zhang, D.-Y., Hu, M.-Z., Zhang, H.-L., & Kang, T.-B. (2005). A network traffic model based on measurement.* Conference paper *(details not specified)*.
2. Aldhyani, T. H. H., & Joshi, M. R. (2016, March). An integrated model for prediction of loading packets in network traffic. In *Proceedings of ICTCS* (pp. 654–660). ACM. https://doi.org/10.1145/2905055.2905236 .
3. Aldhyani, T. H. H., & Joshi, M. R. (2016, March). An integrated model for prediction of loading packets in network traffic. In *Proceedings of ICTCS* (pp. 654–660). ACM. https://doi.org/10.1145/2905055.2905236 .
4. Adekitan, A. I., Abolade, J., & Shobayo, O. (2019). Data mining approach for predicting the daily Internet data traffic of a smart university. *Journal of Big Data, 6*(1), 11. https://doi.org/10.1186/s40537-019-0176-5 .
5. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
6. Huang, C.-W., Chiang, C.-T., & Li, Q. (2018). A study of deep learning networks on mobile traffic forecasting. *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 1–5. https://doi.org/10.1109/VTCFall.2018.8690723.
7. Cortez, P., Rio, M., Rocha, M., & Sousa, P. (2006, July). Internet traffic forecasting using neural networks. In *IJCNN 2006* (pp. 2635–2642). IEEE. https://doi.org/10.1109/IJCNN.2006.247201.
8. Belhaj, S., & Tagina, M. (2009). Modeling and prediction of the Internet end-to-end delay using recurrent neural networks. *Journal of Networks, 4*(6), 528–535.

9. Haviluddin, & Alfred, R. (2014). Forecasting network activities using ARIMA method. *Journal of Advances in Computer Networks, 2*(3), 173–179. https://doi.org/10.7763/JACN.2014.V2.106 .
10. *Oliveira, T. P., Barbar, J. S., & Soares, A. S. (2016). Computer network traffic prediction: A comparison between traditional and deep learning neural networks.* International Journal of Big Data Intelligence, 3*(1), 28–37. https://doi.org/10.1504/IJBDI.2016.073903* .
11. Azzouni, A., & Pujolle, G. (2017). A long short-term memory recurrent neural network framework for network traffic matrix prediction. *arXiv.* https://arxiv.org/abs/1705.05690.
12. Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. *SIGIR '18*, 95–104. https://doi.org/10.1145/3209978.3210006.
13. *Wu, K., Chen, Z., & Li, W. (2018). A novel network intrusion detection model based on convolutional neural networks.* Security and Communication Networks, 2018*, 2513401. https://doi.org/10.1155/2018/2513401*.
14. Li, M., Wang, Y., Wang, Z., & Zheng, H. (2020). LA-ResNet: A spatio-temporal convolutional network with attention for wireless network traffic prediction. *IEEE Access, 8*, 215514–215522. https://doi.org/10.1109/ACCESS.2020.3039925 .
15. Aldhyani, T. H. H., Alrasheedi, M., Alqarni, A. A., Alzahrani, M. Y., & Bamhdi, A. M. (2020). Intelligent hybrid model to enhance time series models for predicting network traffic. *IEEE Access, 8*, 130431–130451. https://doi.org/10.1109/ACCESS.2020.3009169 .

.