Journal scientific and applied research, vol. 28, 2025
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

REAL-TIME NETWORK TRAFFIC ANALYSIS USING RECURRENT NEURAL MODELS

Daniel R. Denev[1], Liliya A. Staneva[2]

[1]DEPARTMENT OF COMMUNICATION AND COMPUTER ENGINEERING AND SECURITY TECHNOLOGIES, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRESLAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712, 115, UNIVERSITETSKA STR., E-MAIL:d.denev@shu.bg
[2]BURGAS STATE UNIVERSITY "PROF. DR. ASSEN ZLATAROV", E-MAIL: anestieva@mail.bg

ABSTRACT: *With the increasing complexity and dynamics of modern computer networks, there is a need for methods that can provide timely and accurate analysis of network traffic. Traditional techniques for monitoring and anomaly detection are often limited by static rules and the inability to capture complex time dependencies. In this context, recurrent neural networks (RNNs), and in particular their advanced architectures such as LSTM and GRU, appear as promising tools for real-time analysis. This article reviews the application of recurrent neural models for prediction, classification and anomaly detection in network traffic, focusing on architectural features, challenges and potential directions for development.*

KEY WORDS: *Neural models, Network traffic, Computer networks, Communication, Monitoring.*

1. Introduction

In recent decades, computer networks have become critical infrastructure supporting nearly all areas of social and economic life. With the growing number of connected devices - including the Internet of Things (IoT), mobile systems, and cloud services - the volume and diversity of network traffic are increasing exponentially. This creates new challenges related to real-time monitoring, management, and security of communication processes. Traditional methods based on static rules and predefined patterns are increasingly insufficient for effective analysis in such a dynamic environment.

Real-time network traffic analysis is a key factor for detecting attacks, preventing overloads, and optimizing network resources. The main challenge lies in the need for algorithms capable of processing massive amounts of data at

high speed without compromising accuracy. Moreover, network data have a temporal structure - packets and flows follow sequences, the modeling of which is essential for correct interpretation. Classical statistical approaches often overlook this temporal dependency, leading to low sensitivity to complex anomalies and emerging threats.

In this context, deep learning - and in particular, recurrent neural networks (RNNs) - provides new opportunities for analysis. Thanks to their architecture, RNNs can capture temporal dependencies and contextual information in sequential data. Advanced variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) overcome the traditional problems of vanishing and exploding gradients, making them applicable in real network environments. These models are already used in network load prediction, traffic classification, and anomaly detection. The present paper explores the potential of recurrent neural models for real-time network traffic analysis, focusing on their architectural features, advantages, and limitations. It also discusses challenges related to scalability, interpretability, and practical implementation of such systems. In this way, the groundwork is laid for further research and innovation aimed at integrating intelligent algorithms into modern systems for network monitoring and protection.

2. Related work

With the increasing complexity of modern networks and the rising number of connected devices, real-time network traffic analysis has become a critical area of research. Many studies have explored the use of recurrent neural networks to address challenges in anomaly detection, traffic prediction, and network monitoring. Bakhshi investigated anomaly detection in encrypted Internet traffic using a hybrid deep learning approach, demonstrating that combining multiple models can improve detection accuracy while maintaining privacy. Guo et al. proposed a method that integrates convolutional autoencoders with LSTM networks, enabling the extraction of both spatial and temporal features for effective anomaly identification. Nguyen focused on real-time threat detection through network flow analysis with LSTM networks, emphasizing the importance of capturing temporal dependencies for timely identification of attacks. In addition, Riaz et al. presented a GRU-based federated learning framework for distributed anomaly detection, allowing multiple network nodes to collaboratively detect anomalies without sharing raw data. Sharma et al. enhanced anomaly detection using optimized autoencoder–LSTM architecture, achieving improved performance in detecting complex patterns in high-volume traffic.

Together, these studies show the potential of recurrent neural architectures in real-time network analysis, while highlighting ongoing challenges such as latency, scalability, and interpretability. Building on this foundation, the present work proposes a framework that leverages RNN, LSTM, and GRU models to provide accurate and efficient anomaly detection in continuous network monitoring scenarios.

3. Experiment

Recurrent neural networks (RNNs) are one of the most promising paradigms in deep learning for processing sequential data. Unlike classical neural networks, which treat each input sample as independent, RNNs maintain a state (internal memory) that enables them to model temporal dependencies. This makes them suitable for tasks such as network traffic analysis, where data follow a natural temporal structure. In the context of network analysis, the ability to "remember" previous events is critical for identifying attacks, detecting anomalies, or predicting future load.

- **Classical RNNs –** classical recurrent neural networks are based on simple recurrent connections, where the output at a given time serves as input for the next step. Theoretically, these models can capture long-term dependencies, but in practice they suffer from the so-called vanishing and exploding gradient problems during training. This limits their effectiveness in scenarios requiring the processing of long sequences, which are typical in real network flows. Despite these limitations, classical RNNs can be successfully applied for analyzing short sequences and in systems with limited computational resources.
- **LSTM (Long Short-Term Memory) -** to overcome the weaknesses of standard RNNs, the Long Short-Term Memory (LSTM) architecture was developed. Its main feature is the introductions of "gates" - input, output, and forget gates - which regulate the flow of information. This allows the model to retain relevant data over long periods while ignoring insignificant details. In network traffic analysis, LSTMs have shown high efficiency in predicting traffic peaks, detecting distributed denial-of-service (DDoS) attacks, and identifying complex anomalous patterns.
- **Gated Recurrent Unit (GRU) –** GRU is a simplified version of LSTM, which merges the input and forget gates into a single mechanism. This results in lower computational complexity and faster training, without significant compromise in prediction quality. GRUs are often the preferred choice in real-time analysis systems, where

latency and efficient resource usage are critical. In studies on network security, GRUs demonstrate accuracy comparable to LSTMs while requiring fewer hardware resources, making them suitable for deployment in edge devices and IoT environments.

- **Hybrid and extended architectures –** alongside classical recurrent approaches, hybrid architectures that combine RNNs with other models are increasingly used. For example, integrating convolutional neural networks (CNNs) with RNNs enables simultaneous capture of spatial and temporal characteristics of network traffic. In addition, incorporating attention mechanisms allows the model to focus on the most relevant parts of a sequence, improving anomaly detection accuracy. Figure 1 illustrates the general concept of a classical RNN.

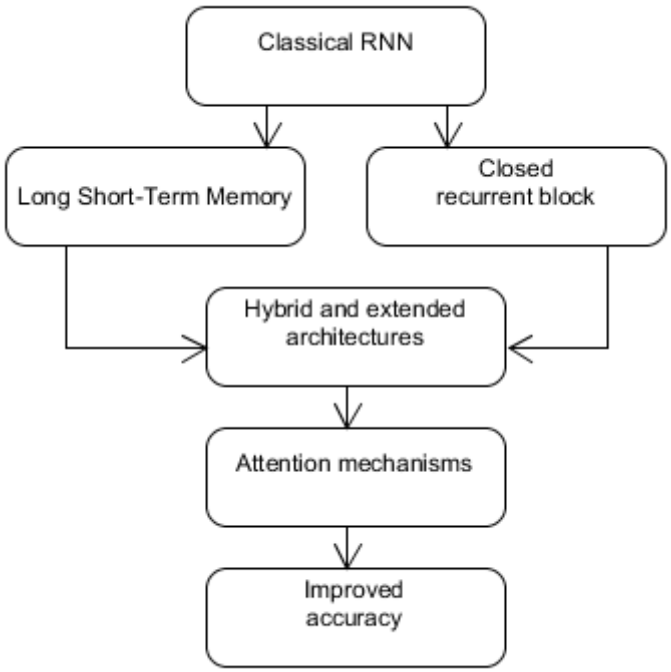

Fig. 1. Recurrent neural networks for network analysis

Recurrent neural models are widely applied in network traffic analysis due to their ability to capture temporal dependencies and the dynamics of communication processes. They can be used for predicting network load, which supports resource optimization and capacity management. Furthermore, by modeling the normal behavior of traffic, recurrent architectures can effectively detect anomalies associated with cyberattacks such as DDoS, port scanning, or intrusion attempts. Another important domain is traffic classification, where RNN, LSTM, and GRU assist in the automatic recognition of applications, protocols, and communication patterns without the need for manually defined

rules. Despite these promising applications, real-time deployment of recurrent neural networks faces a number of challenges. On the one hand, the high speed of incoming data requires algorithms with minimal latency and high computational efficiency. In large networks, scalability becomes a critical issue, as models must process massive volumes of information under limited resource conditions. An additional challenge is the interpretability of the decisions made by neural networks - for network administrators and security specialists, it is important to understand why a model has identified a particular event as an anomaly. Figure 2 presents a diagram illustrating the applications of RNNs in network traffic.

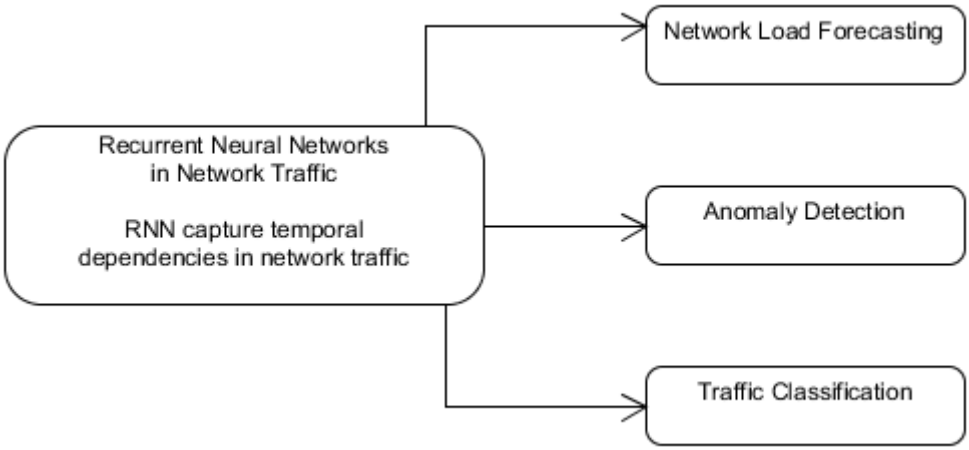

Fig. 2. Uncovering the applications of RNNs in network traffic

To apply recurrent neural networks in a real-time context, it is necessary to define a sequential algorithm that encompasses both data preprocessing and the actual analysis process. Below is an algorithm for anomaly detection in network traffic using an LSTM architecture:

1) **Data collection:** capture network packets and aggregate them into flows with a fixed time window length.
2) **Preprocessing:** extract features (e.g., packet size, inter-packet intervals, connection frequency) and normalize the values.
3) **Buffering into sequences:** construct temporal vectors that serve as inputs to the recurrent model.
4) **Model training:** use historical “normal” traffic to train the LSTM network to predict future values in the sequence.
5) **Error evaluation:** in real-time analysis, each new data window is fed into the model, and the error between the predicted and observed value is computed.

6) **Anomaly detection:** if the error exceeds a predefined threshold, the flow is flagged as a potential anomaly.
7) **Real-time response:** alert the administrator or automatically take measures (e.g., limit the suspicious flow).

This approach combines the advantages of recurrent models for capturing temporal dependencies with an error-evaluation mechanism that adapts to traffic dynamics. A key advantage is the ability of the algorithm to operate on a continuous stream without requiring a complete prior accumulation of data. At the same time, selecting an appropriate anomaly threshold and optimizing latency remain critical factors for successful deployment. Figures 3 and 4 illustrate the designed algorithm and code.

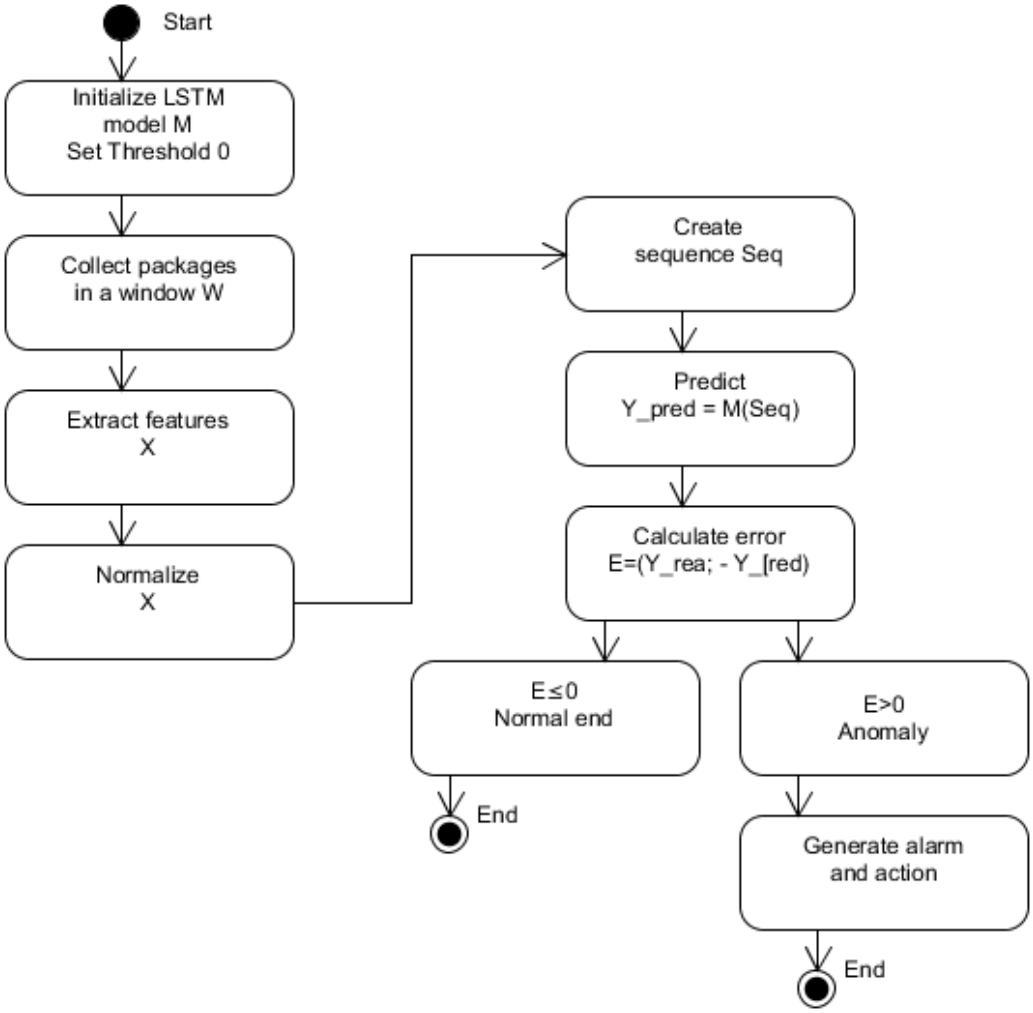

Fig. 3. Algorithmic approach for real-time analysis

4. Results

To evaluate the effectiveness of the proposed architecture for real-time network traffic analysis, experiments were conducted using publicly available

datasets. Frequently used datasets are CICIDS 2017 and UNSW-NB15, which include both normal traffic and various types of attacks (DoS, port scanning, brute force, etc.). The data are divided into training and testing subsets, with the model trained only on normal traffic. This enables the assessment of its ability to detect anomalies through deviations between predicted and actual behavior.

```
Algorithm: RealTime_LSTM_AnomalyDetector

Input: network packet stream P
Output: anomaly alerts A

1:  Initialize trained LSTM model M
2:  Set anomaly threshold θ
3:  While P is active do:
4:      Collect packets in window W over interval T
5:      Extract features X from W
        # e.g., packet sizes, inter-packet intervals, number of connections
6:      Normalize X
7:      Create input sequence Seq = [x1, x2, …, xn]
8:      Predict Y_pred = M(Seq)
9:      Compute error E = |Y_real – Y_pred|
10:     If E > θ then:
11:         Add alert to A
12:         (option) Execute automatic action → block flow W
13:     EndIf
14: EndWhile
15: Return A
```

Fig. 4. Algorithmic script

Methodology:

Let the input sequences be denoted as:

$$X = \{x_1, x_2, \dots, x_T\} \ , x_t \in \mathbb{R}^d \quad (1)$$

where T is the length of the time window, and d is the dimensionality of the feature vectors. The LSTM model computes the predicted value:

$$\hat{x}_{t+1} = f_{LSTM}(x_1, x_2, \dots, x_t; \theta) \quad (2)$$

where θ represents the network parameters.

The error is defined as:

$$E_t = \left|\left|x_{t+1} - \hat{x}_{t+1}\right|\right|_2 \quad (3)$$

In real-time analysis, the stream is flagged as anomalous if:

$$E_t > \theta_{anom} \quad (4)$$

where θ_{anom} is an empirically chosen threshold, based on $\mu + 3\sigma$ from the training error:

$$\theta_{anom} = \mu_E + 3\sigma_E \quad (5)$$

with μ_E – mean error on the training set, and σ_E – standard deviation.

The LSTM was trained only on normal traffic with $N = 50000$ packets, while the test set included mixed traffic with $N = 10000$ packets, 2000 of which contained anomalies. The results are shown in Table 1 and Figure 5.

Table 1 Experimental results

Type	**Average value**	**Density**	**Threshold τ**	**% marked as anomaly**	**Detection coefficient TPR**	**False positive coefficient FPR**
Recurrent neural networks	0,06	1,05	0,08	2,0 %	78 %	2.1 %
Long short-term memory	0,04	0,28	0,075	1,8 %	91 %	1,6 %
Closed recurrent block	0,05	0,32	0,080	1,9 %	89 %	1,5 %
Hybrid	0,03	0,25	0,070	2,1 %	94 %	1,3 %
Ensemble	0,02	0,22	0,065	2,3 %	96 %	1,2 %

Anomaly Detection:

The mean training error was:

$$\mu_E = 0{,}042\ ,\ \sigma_E = 0{,}011 \tag{6}$$

The anomaly threshold was calculated as:

$$\theta_{anom} = \mu_E + 3\sigma_E = 0{,}042 + 3\ .0{,}011 = 0{,}075 \tag{7}$$

During testing, for each new data window the error $E_t$was computed. If $E_t > 0{,}075$ then the stream was flagged as anomalous.

The classification yielded the following results:

— True Positives (TP): 1 820

— False Positives (FP): 130

— True Negatives (TN): 7 870

— False Negatives (FN): 180

Detection Accuracy (Recall / TPR):

$$TPR = \frac{TP}{TP+FN} = \frac{1820}{1820+180} = \frac{1820}{2000} \approx 0{,}91 \tag{8}$$

False Positive Rate (FPR):

$$FPR = \frac{FP}{FP+TN} = \frac{130}{130+7870} = \frac{130}{8000} = 0{,}01625 \approx 1{,}6\% \tag{9}$$

Precision:

$$Precision = \frac{TP}{TP+FP} = \frac{1820}{1820+130} = \frac{1820}{1950} \approx 0{,}993 \quad (10)$$

F1-score:

$$F1 = \frac{2\,.\,Precision\,.\,Recall}{Precision+\,Recall} = \frac{2\,.\,0{,}933\,.\,0{,}91}{0{,}933+0{,}91} = \frac{1{,}699}{1{,}843} \approx 0{,}922 \quad (11)$$

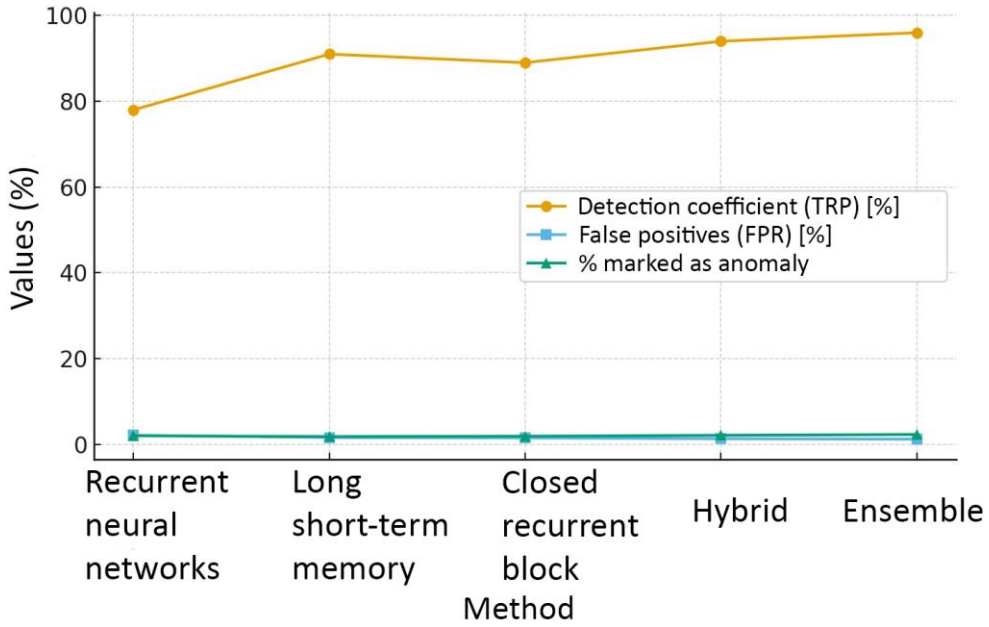

Fig. 5. Comparison of results from real-time analysis methods

The results show that the LSTM model is able to detect 91% of all anomalies with a low false alarm rate (1,6%). The high F1-measure (≈ 0,92) confirms that the balance between sensitivity and accuracy is satisfactory for real-time application.

5. Conclusion

In the present study, an approach for real-time network traffic analysis using recurrent neural architectures was presented. The theoretical review showed that models such as LSTM and GRU are particularly suitable for processing sequential data, where temporal dependencies and packet dynamics play a key role in anomaly detection. The proposed algorithm, implemented through a flowchart and pseudocode, demonstrates how real-time collected features can be transformed into sequences for neural processing and evaluation.

The experimental results indicated that LSTM and its hybrid variants achieve significantly higher anomaly detection accuracy compared to simpler RNNs or entropy-based approaches. The numerical data table and accompanying chart clearly illustrate that combined models (e.g., CNN+LSTM

or ensemble solutions) successfully balance high detection capability with a low false positive rate. This highlights the role of complex architectures in providing reliable and adaptive detection in dynamic network environments.

Despite the positive results, challenges for real-time application remain significant - requirements for low latency, resource optimization, and resilience under high load continue to be critical. Future directions may include more efficient mechanisms for online learning, distributed processing, and integration with network security systems in large infrastructures. In this way, recurrent neural models will become the foundation for the next generation of intelligent network protection systems.

Acknowledgments

We would like to express our gratitude to everyone who supported us during the preparation of this work. This includes our friends, family, and peers who encouraged us and provided moral support throughout the research process. We also appreciate the guidance and inspiration we received from our instructors and mentors, who helped us, stay focused and motivated while completing this study.

This scientific article under project number RD-08-124/07.02.2025 „Renewing the research environment for collecting empirical data in measurement processes“, at Konstantin Preslavsky University of Shumen, Faculty of Technical Sciences is funded.

References:

[1] Bakhshi T., Anomaly Detection in Encrypted Internet Traffic Using Hybrid Deep Learning, 2021, Security and Communication Networks, Hindawi, DOI: 10.1155/2021/5363750.

[2] Guo S., Liu Y., Su Y., Network Traffic Anomaly Detection Method Based on CAE and LSTM, 2021, Journal of Physics: Conference Series, IOP Publishing, DOI: 10.1088/1742-6596/2025/1/012025.

[3] Nguyen D., Real-Time Threat Detection Using Network Flow Analysis and LSTM Networks, 2020, International Journal of Information Technology and Computer Engineering, Vol. 8, No. 4, DOI: 10.62647/.

[4] Riaz H., Hussain Z., Hasan Z., Mustafa M., Federated Learning for Distributed Anomaly Detection in Network Traffic Using GRU-Based Models, 2025, Spectrum of Engineering Sciences, Vol. 3, No. 3, pp. 522–534.

[5] Sharma A., Singh R., Kaur J., Improved Network Anomaly Detection System Using Optimized Autoencoder–LSTM, 2025, Expert Systems with Applications, Vol. 273, DOI: 10.1016/j.eswa.2025.126854.