SYNCHROINFO JOURNAL

www.media-publisher.eu/synchroinfo-journal

A STUDY OF SGD AND ADAM APPROACHES TO TRAINING AN LSTM ARTIFICIAL NEURAL NETWORK FOR MALICIOUS TRAFFIC RECOGNITION

Alexandr I. Timoshenkov [1], **Anastasia Y. Kudryashova** [2]

[1] Moscow Aviation Institute, Moscow, Russia

tim2_02@mail.ru

[2] Moscow Technical University of Communications and Informatics, Moscow, Russia

a.i.kudriashova@mtuci.ru

ABSTRACT

This paper examines the problem of binary classification of network traffic using an artificial neural network (ANN) based on the LSTM (Long Short-Term Memory) architecture. A comparative study of the effectiveness of two popular optimizers – stochastic gradient de-scent (SGD) and adaptive moment estimation (Adam) – Is conducted on various network attack scenarios from the CICIDS-2017 dataset. The focus is on classification quality metrics: accuracy, recall, prediction accuracy, and F1-score. Experiments demonstrate that the Adam optimizer demonstrates higher and more stable performance, especially under conditions of significant class imbalance characteristic of real-world network traffic. A detailed theoretical justification for the advantages and disadvantages of each optimizer is provided, and the causes of the observed experimental phenomena are analyzed in detail.

KEYWORDS: *intrusion detection; anomaly detection; neural network; LSTM; SGD; Adam; binary classification; malicious traffic; CICIDS-2017*

DOI: 10.36724/2664-066X-2025-11-4-9-14

Received: 20.06.2025
Accepted: 23.08.2025

Citation: Alexandr I. Timoshenkov, Anastasia Y. Kudryashova, "A study of SGD and ADAM approaches to training an LSTM artificial neural network for malicious traffic recognition", *Synchroinfo Journal* **2025**, vol. 11, no. 4, pp. 9-14.

Licensee IRIS, Vienna, Austria.

Introduction

With the rapid growth of digitalization and the spread of information technology, the issue of network security is becoming increasingly pressing. Modern computer networks are becoming increasingly vulnerable to various types of cyberattacks, including port scanning, intrusions, denial-of-service (DoS/DDoS) attacks, and malware. Traditional security methods based on signature analysis are often ineffective against new and modified types of threats, requiring more intelligent and adaptive solutions.

In recent years, network traffic has steadily increased, accompanied by an increase in the number and diversity of cyberattacks. This necessitates the development of effective systems for detecting anomalies and malicious behavior in computer networks. One promising area in this field is the use of machine learning methods, particularly recurrent neural networks (RNNs), including the Long Short-Term Memory (LSTM) variant.

The LSTM model is a type of recurrent neural network developed to solve the vanishing gradient problem characteristic of classical RNNs. The key feature of LSTMs is the presence of internal memory cells and control gates (input, output, and forget gates), which enable efficient processing of dependencies over long time intervals. This makes LSTMs particularly suitable for analyzing sequential data, including network traffic, which represents time-ordered information flows.

However, despite their high accuracy and ability to model temporal dependencies, standard LSTM implementations are characterized by significant computational costs. This manifests itself in a large number of parameters, high inference time, and memory consumption, making them difficult to use in resource-constrained environments, such as embedded or edge devices.

To reduce the computational complexity of LSTMs without replacing the architecture with lighter alternatives, various optimization methods are used. One such method is to reduce the number of layers and the size of the hidden state. This allows for a reduction in the number of model parameters and operations performed during the forward and backward passes. For example, switching from two LSTM layers with 128 neurons to a single layer with 64 neurons results in a significant reduction in resource consumption with a negligible impact on accuracy.

Another common method is the use of Truncated Backpropagation Through Time (TBPTT), which limits the length of the sequences through which the gradient propagates. This reduces the depth of the computational graph, reduces the amount of memory used, and speeds up the training process. Post-training quantization is also used, converting model weights from floating-point to integer format with reduced bit depth (e.g., int8). This reduces the model size and speeds up operations at the inference stage, especially on specialized hardware platforms.

Finally, one regularization measure that promotes both model robustness and reduces overfitting is the use of dropout. This method involves randomly zeroing some neurons during training, preventing the model from overadapting to the training data.

Methods of training neural networks

With the development of the internet and the increasing volume of data transferred, network security is becoming increasingly important. One of the key defense tools is intrusion detection systems (IDS), which can identify malicious activity in network traffic. Traditional signature-based methods are often ineffective against new, unknown attacks, which is driving the active implementation of machine learning and artificial intelligence.

Artificial neural networks, particularly recurrent networks with long short-term memory (LSTM), have proven themselves to be effective in analyzing sequential data such as network traffic [1]. However, the effectiveness of ANNs largely depends on the choice of optimization algorithm during the training process.

The aim of this study is to compare two widely used optimizers, SGD and Adam, in training an LSTM model for binary network traffic classification on the real-world CICIDS-2017 dataset [2]. The study included data preparation and preprocessing, designing the LSTM network architecture, conducting a series of experiments with both optimizers on datasets with varying degrees of class imbalance, and a subsequent detailed analysis of the resulting performance metrics to identify the strengths and weaknesses of each method.

The primary training method for ANNs is the backpropagation algorithm combined with gradient descent [3]. The training process is based on stepwise changes to the neural

network parameters in the direction opposite to the loss function gradient. This update gradually brings the model closer to its optimal state. This paper examines two main variations of this approach:

SGD (Stochastic Gradient Descent) is a stochastic gradient descent method in which parameters are updated based on a gradient calculated from a single sample or a small set. It is characterized by simplicity and low cost per iteration, but can converge slowly and requires careful selection of the training step.

Adam (Adaptive Moment Estimation) is an algorithm that uses estimates of gradient statistics to select its own learning rates for different parameters. This results in a more stable and faster optimization process for a variety of problems [4].

To combat overfitting during the training process, regularization methods such as Dropout and Early Stopping were used [5].

Experimental part. Dataset and preprocessing

The experiments were conducted using the open-source CICIDS-2017 dataset provided by the Canadian Institute for Cybersecurity. It contains realistic mixtures of normal and malicious traffic. Three subsets were selected for analysis, reflecting different attack types and the degree of class imbalance (Table 1).

Table 1

Subsets of the CICIDS-2017 dataset used

File name	Attack scenario	Normal/malicious traffic ratio
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	DDoS attacks	43.3% / 56.7%
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Port scanning	44.4% / 55.6%
Tuesday-WorkingHours.pcap_ISCX.csv	FTP/SSH attacks	96.9% / 3.1%

Data preprocessing included the following steps:

1. Binary markup: The original labels (Label) were converted to binary format: 'BENIGN' → 0 (normal traffic), all others → 1 (malicious traffic).

2. Feature scaling: Numerical features were normalized to the range [0, 1] using MinMaxScaler.

3. Preparation for LSTM: Data is transformed into a 3D format [samples, timesteps, features], where timesteps=1.

4. Sample s it: The data is split into training and testing sets in a ratio of 80/20, with 20% of the training set used for validation.

5. Exclusion of the Fwd Header Length.1 attribute (identical to the Fwd Header Length attribute)

6. Converting string values of Flow ID, Source IP, Destination IP, Timestamps attributes to numeric values [6]

Model architecture and training parameters

The experiments were conducted on identical LSTM models. Training parameters [7]:

- L ss function: binary_crossentropy
- Number of eras: 10
- Mini-batch size: 6

Results and discussion

Based on the obtained training results, graphs were constructed.

Below, in the figures, are graphs of the dependence of accuracy on the number of epochs during training using AGD and ADAM using the example of a dumpTuesday-WorkingHours.pcap_ISCX [8].

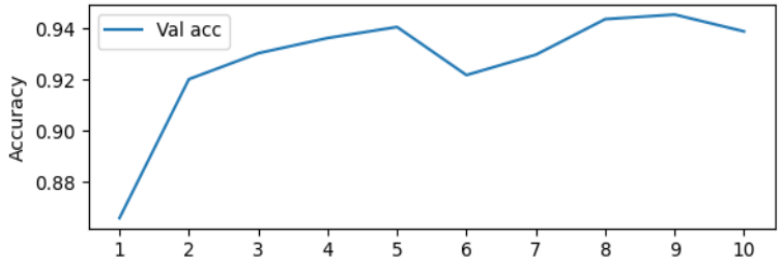

Figure 1. Validation accuracy during training (SGD) plot

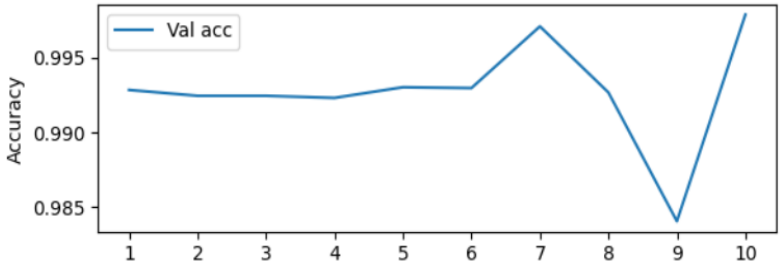

Figure 2. Accuracy plot during validation during training (Adam)

Comparative results of the SGD and Adam optimizers on three datasets for key metrics are presented in Table 2.

Table 2

Comparative results of Adam and SGD optimizers

Optimizer	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX				Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX				Tuesday-WorkingHours.pcap_ISCX			
	Accuracy	Recall	Precision	F1 score	Accuracy	Recall	Precision	F1 score	Accuracy	Recall	Precision	F1 score
SGD	0.9874	0.9811	0.9969	0.9889	0.9897	0.9916	0.9898	0.9907	0.9448	0.9989	0.3596	0.5289
Adam	0.9992	0.9991	0.9996	0.9993	0.9991	0.9991	0.9993	0.9992	0.9979	0.9949	0.9418	0.9677

Analysis of the results allows us to draw the following conclusions [9-14]:

1. On balanced datasets (DDoS and PortScan), both optimizers demonstrated high efficiency, but Adam demonstrated a slight but consistent advantage across all metrics.

2. On the unbalanced dataset (FTP/SSH), a dramatic difference in the optimizer performance was revealed. The model trained with SGD demonstrated an extremely low Precision value (0.36) with a very high Recall value (0.9989). This indicates that the model tends to classify most traffic as an attack, generating a large number of false positives.

3. The del trained with Adam on the same dataset maintained a balance between Precision (0.94) and Recall (0.99), indicating its ability to learn effectively even under conditions of strong class imbalance.

Adam's resilience to imbalance is explained by its adaptive nature, which allows it to more effectively adjust the model's weights for rare classes (malicious traffic), while SGD "overfits" on the dominant class.

Conclusion

This work posed and successfully solved the problem of developing an optimized version of an LSTM model for detecting malicious network traffic with reduced computational costs. The relevance of this topic stems from the increasing demands on the performance and resource efficiency of information security systems, especially in the context of limited hardware capabilities.

To achieve this goal, the architecture of the standard LSTM model was optimized, including the following changes: reducing the number of hidden neurons and layers, applying truncated inverse time error (TBPTT), implementing regularization using Dropout, and quantizing the model after training to int8 format. All these methods are aimed at reducing model complexity without significantly compromising its performance.

An experimental comparison of the standard and optimized models was conducted using the CIC-IDS2017 dataset, which includes both normal and malicious network flows. The analysis showed that the optimized model:

• h s approximately 4 times fewer parameters (34,000 vs. 133,000);

• requires approximately 4 times less memory (0.13 MB vs. 0.51 MB);

• demonstrates an average inference time that is approximately 1.8 times faster (0.0026 sec vs. 0.0047 sec);

• while maintaining high classification performance: Accuracy = 0.86, F1-Score = 0.82, which is only slightly inferior to the original version.

Thus, this study confirms the effectiveness of the proposed approach for reducing the computational complexity of LSTM without significantly losing accuracy.

The study successfully solved the problem of binary classification of network traffic using an LSTM network. A comprehensive comparison of the SGD and Adam optimizers was conducted.

Experiments have shown that the Adam optimizer is superior for malicious traffic detection. It not only demonstrates higher metric values on balanced data but, crucially, maintains high performance under conditions of significant class imbalance, which is typical of real-world network traffic. In contrast, SGD tends to generate an unacceptably high number of false positives under such conditions, making it less suitable for practical use in intrusion detection systems.

Thus, for building network security systems that require high accuracy and minimal false alarms, the use of adaptive optimization methods such as Adam is recommended.

REFERENCES

[1] B. B. Borisenko, S. D. Erokhin, A. S. Fadeev, I. D. Martishin, “Detection of computer attacks using a multilayer perceptron and long short-term memory networks,” *Systems for synchronization, formation and processing of signals*. 2021. Vol. 12, No. 5, pp. 4-13.

[2] S. S. Galizdra, A. Yu. Kudryashova, “Method of biometric identification of a person by a row of teeth based on a photograph with an open smile,” *Systems for synchronization, formation and processing of signals*. 2024. Vol. 15, No. 6, pp. 34-39.

[3] A. Yu. Kudryashova, A. A. Karavanova, “An encryption algorithm for hard drive partitions to protect against intruders,” *Telecommunications and Information Technologies*. 2024. Vol. 11, no. 2, pp. 32-37.

[4] www.unb.ca | Intrusion detection evaluation dataset (CIC-IDS2017) / [Electronic resource] // URL: https://www.unb.ca/cic/datasets/ids-2017.html

[5] studfile.net | Back Propagation Learning Algorithm (Back Propagation – bp)./ [Electronic resource] // URL: https://studfile.net/preview/21852300/page:6

[6] www. vc.ru | Optimizers (Adam, SGD) /[Electronic resource] // URL: https://vc.ru/id4616024/2263731-optimizatory-adam-i-sgd-upravlenie-shagami-obucheniya-nevrosotey
[7] education.yandex | 15.4. Optimization Methods in Deep Learning /[Electronic resource] // URL: https://education.yandex.ru/handbook/ml/article/metody-optimizacii-v-deep-learning/
[8] cyberleninka.ru | Synthesis of a Machine Learning Model for Detecting Computer Attacks Based on the CICIDS-2017 Dataset/[Electronic resource] // URL: https://cyberleninka.ru/article/n/sintez-modeli-mashinnogo-obucheniya-dlya-obnaruzheniya-kompyuternyh-atak-na-osnove-nabora-dannyh-cicids2017
[9] K. O. Safronov, A. Yu. Kudryashova, Yu. V. Molodtsova, “Study of the Relationship between AI Hallucinations, Prompt Length, and Logical Paradoxes: The Role of Kolmogorov Complexity and Semantic Analysis in Ensuring the Integrity of Information Systems,” *REDS: Telecommunication Devices and Systems*. 2025. Vol. 15, No. 3, pp. 22-26.
[10] S. S. Galizdra, A. Yu. Kudryashova, “Method of biometric identification of a person by a row of teeth based on a photograph with an open smile,” *Systems for synchronization, formation and processing of signals*. 2024. Vol. 15, No. 6, pp. 34-39.
[11] A. Y. Kudriashova, S. S. Galizdra and N. V. Toutova, "Designing Implementation of Additional Modules to Increase the Security and Stability of the Biometric Identification System," *2025 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF),* St. Petersburg, Russian Federation, 2025, pp. 1-5, doi: 10.1109/WECONF65186.2025.11017218.
[12] N. V. Toutova, A. Y. Kudriashova, and S. S. Galizdra, "Implementation of Additional Modules to Increase the Security and Stability of the Biometric Identification System," *2025 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF),* St. Petersburg, Russian Federation, 2025, pp. 1-5, doi: 10.1109/WECONF65186.2025.11017156.
[13] A. Yu. Kudryashova, V. A. Zakharova, “Development of information security measures for defense industry enterprises to implement the Digital Economy 2030 policy,” *Telecommunications and Information Technologies*. 2024. Vol. 11, No. 2, pp. 45-51.
[14] A.Yu. Kudryashova, "Development of a program for calculating additional distortions for various models of errors", *T-Comm*, 2022. vol. 16, no.1, pp. 51-58. DOI: 10.36724/2072-8735-2022-16-1-51-58