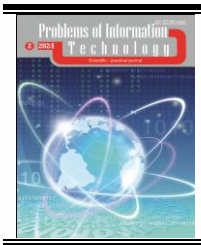

Available online at www.jpit.az

15 (2) 2024

Network cybersecurity incidents multiclassification based on deep learning

Rasim Alguliyev[a], Ramiz Shikhaliyev [b]

[a,b] Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

[a] r.alguliev@gmail.com; [b] shikhramiz61@gmail.com

[a] https://orcid.org/0000-0003-1223-7411; [b] https://orcid.org/0000-0002-8594-6721

ARTICLE INFO

Keywords:

Network cybersecurity incidents
Network cybersecurity incidents multiclassification,
Deep learning model,
CNN-LSTM model
Network traffic classification

ABSTRACT

The rapid increase in network traffic and the growing complexity of cyberattacks have rendered traditional cybersecurity monitoring methods insufficient for effectively detecting and classifying network incidents. To overcome these limitations, we present a deep learning-based approach that utilizes a hybrid architecture, combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models, for the multi-classification of cybersecurity incidents. Our model is trained on the CICIDS2017 dataset, which encompasses a wide range of attack types. The hybrid CNN-LSTM model achieved a classification accuracy of 96.76% and an error rate of 9.34%, showcasing its ability to accurately detect and classify various cybersecurity threats. This approach offers a robust solution for enhancing the detection and classification of network cybersecurity incidents.

1. Introduction

In today's hyperconnected world, the frequency and sophistication of network cybersecurity incidents have grown exponentially. The rapid increase in cyber threats, including malware attacks, Distributed Denial of Service (DDoS), ransomware, and phishing, has created an urgent need for effective mechanisms to detect, classify, and mitigate security incidents (Chen, 2020).

Traditional methods of cybersecurity incident detection, which often rely on rule-based systems or signature detection, are insufficient in dealing with the evolving nature of modern cyberattacks (Juyal et al., 2023). These methods struggle to adapt to novel threats and often generate high false-positive rates, overwhelming security teams with false alarms.

Traditional approaches for detecting and classifying cybersecurity incidents, while effective against known threats, face challenges when dealing with evolving and zero-day attacks. These methods tend to generate high false-positive rates and are unable to adequately handle the vast, complex, and high-dimensional data generated by modern networks. To address these challenges, there has been a shift toward deep learning (DL) techniques for cybersecurity, which have demonstrated superior capabilities in learning from large datasets and uncovering hidden patterns of malicious activities (Vinayakumar et al., 2019; Juyal et al., 2023).

Among advanced DL architectures, Convolutional Neural Networks (CNNs) (Tripathy and Singh, 2021) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have shown particular promise. CNNs excel at capturing spatial relationships in data, such as those found in network traffic features, while LSTMs are powerful for handling temporal dependencies, making them well-suited for sequential data like time-series analysis of network logs or traffic flows. When combined, these models can harness the strengths of both architectures, enabling more

Received 26 February 2024, Received in revised form 25 April 2024, Accepted 13 May 2024
http://doi.org/10.25045/jpit.v15.i2.03

effective multi-class classification of cybersecurity incidents. This hybrid CNN-LSTM model captures both the spatial features of network data and its temporal dynamics, which is crucial for accurately detecting and categorizing a wide range of attacks.

This article introduces a hybrid CNN-LSTM model for the multi-class classification of network cybersecurity incidents. The proposed model aims to accurately classify network incidents into multiple categories, such as malware, phishing, or denial-of-service attacks, by leveraging the complementary strengths of CNNs and LSTMs.

By combining the spatial learning capabilities of CNNs with the temporal modeling power of LSTMs, this research provides a novel approach to the multi-classification of cybersecurity incidents, offering improved detection accuracy and timely responses to a broad spectrum of network-based threats. This hybrid architecture holds great promise for advancing automated cybersecurity systems and enhancing the resilience of digital infrastructures against evolving cyberattacks.

The remainder of this article is organized as follows: Section 2 reviews the relevant literature. Section 3 describes the DL-based method for multi-classifying network cybersecurity incidents. Section 4 explains the proposed network cybersecurity incidents multi-classification model. Section 5 discusses the dataset and experimental setup. Section 6 presents and discusses the experimental results. Finally, the conclusion highlights potential directions for future research.

2. Related works

The use of DL techniques in network cybersecurity has attracted considerable attention in recent years, as traditional methods like signature-based detection and rule-based systems have proven inadequate in addressing the evolving nature of cyber threats and the complexity of modern networks.

DL has proven effective in detecting and classifying various types of cyber incidents due to its ability to automatically learn patterns from large datasets. Several studies have demonstrated the use of deep learning models, such as CNNs, Recurrent Neural Networks (RNNs), and LSTM networks, for cybersecurity applications. CNNs have been widely applied to network traffic data to capture spatial relationships between features, such as in intrusion detection systems.

Xiao et al. (2019) used CNNs to classify different types of network attacks by analyzing packet-level features, showing improved detection rates compared to traditional approaches.

LSTMs, on the other hand, have shown particular strength in handling time-series data, making them suitable for analyzing sequential network logs and traffic flows.

Zhang et al. (2019) utilized LSTM networks to detect DDoS attacks by learning temporal features in network traffic. LSTMs' ability to capture long-term dependencies in sequential data is especially valuable for detecting attacks that unfold over time, such as advanced persistent threats (APTs).

Combining CNNs and LSTMs has emerged as a powerful strategy to leverage the strengths of both architectures for cyber incident detection and classification. CNNs are adept at extracting spatial features, while LSTMs excel at modeling temporal dependencies. Several recent studies have explored hybrid CNN-LSTM models for cybersecurity tasks, finding that these architectures outperform single-model approaches in various domains.

Halbouni et al. (2022) proposed a hybrid CNN-LSTM model for intrusion detection, where CNNs were used to extract local spatial features from network traffic data, and LSTMs captured the temporal dependencies. This approach significantly improved the detection of complex attack patterns compared to standalone CNN or LSTM models.

Alzahrani et al. (2022) applied a hybrid CNN-LSTM model to detect botnet attacks, achieving higher classification accuracy and lower false-positive rates than traditional machine learning models like Random Forest and SVM.

The task of multi-class classification in cybersecurity, which involves classifying network traffic or events into multiple categories (e.g., malware types, phishing, or DDoS attacks), poses unique challenges due to the high dimensionality and diversity of attack types. Several studies have explored deep learning models for multi-class classification.

Despite the promising results of hybrid CNN-LSTM models, several challenges remain. The high computational cost of training such models, especially on large-scale datasets, is one key concern. Moreover, imbalanced datasets, where benign traffic vastly outnumbers attack data, can lead to biased models. Recent efforts, such as using data augmentation techniques and more balanced sampling, have been proposed to address these challenges (Otokwala et al., 2021; Medvedieva et al., 2024).

In addition, there is ongoing research into enhancing the generalization capabilities of deep learning models for cybersecurity, particularly in dealing with zero-day attacks and evolving threat

landscapes (Buczak and Guven, 2016). Transfer learning and semi-supervised learning approaches (Mbona and Eloff, 2022) have been explored as ways to mitigate these issues, allowing models to adapt to unseen or underrepresented classes of incidents.

We develop a novel hybrid network cybersecurity incidents multi-classification model that leverages the strengths of both CNNs and LSTMs for effective spatiotemporal feature extraction. To enhance feature processing, we introduce a novel preprocessing technique to improve the extraction of relevant features from raw network traffic data. Trained and tested on the CICIDS2017 dataset, the proposed hybrid CNN-LSTM model demonstrates high classification accuracy and operational speed.

3. Method for network cybersecurity incidents multi-classification based on DL

The proposed approach for network cybersecurity incidents multi-classification leverages a hybrid deep learning architecture

Problem statement. The task of network cybersecurity incidents multi-classification involves the identification and classification of various types of traffic data to detect potential cybersecurity incidents. The goal is to develop a deep learning-based approach that effectively classifies network traffic data into one of the predefined classes, each representing a different type of cybersecurity incident or benign traffic. The primary objective is to extract and leverage spatiotemporal features from network traffic to improve the accuracy and robustness of the classification process.

Problem definition. Given a dataset D consisting of network traffic records, where each record x_i is associated with a label $y_i \in \{1, 2, \dots, n\}$ representing the class of the traffic, the goal is to design a model $f: X \to Y$ that maps the input features X to the correct class labels Y.

The proposed approach includes gathering data from cybersecurity incidents in computer networks, extracting spatiotemporal features, and training DL model (Figure 1). The collected data analysis is a critical step in the cybersecurity monitoring of CN to identify security threats.

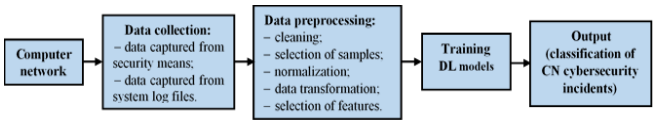

Fig. 1. Network cybersecurity incidents multi-classification based on DL

The proposed method consists of several components. The data collection component gathers data from computer networks security systems or system log files containing information about various computer networks cybersecurity events.

Data preprocessing includes data cleaning, sample selection, normalization, transformation, and feature selection. In the DL model training component, DL models are trained. The output component classifies computer networks cybersecurity incidents.

4. Network cybersecurity incidents multi-classification model

The proposed model comprises multiple layers (Figure 2). We used the architecture of CNN-LSTM adapted for one-dimensional multi-classification (Figure 3). This model integrates CNN layers to extract spatial features and LSTM layers to capture temporal dependencies.

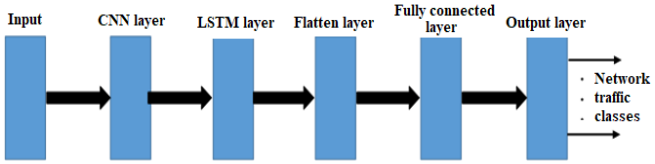

Fig. 2. Network cybersecurity incidents multi-classification model

The input to the model will be a data sequence with a single feature per time step.

The CNN layer consists of a Conv1D layer and a MaxPooling layer. The Conv1D layer conducts convolutions along the temporal dimension of the input data to capture spatial patterns. This layer employs 64 filters (kernels) with a kernel size of 3 to learn distinct features from the input. ReLU activation functions are employed to introduce non-linearities. Following this, the MaxPooling layer performs down-sampling along the temporal dimension by selecting the maximum value within a pool size of 2. The CNN layer can be described as follows:

Convolution with a 1D kernel is:

$$y_{t,j} = ReLU\left(\sum_{i=0}^{k-1} x_{t+1} w_{i,j} + b_j\right), \quad (1)$$

where x is the input sequence, w are the weights (filters), b are the biases, k is the kernel size, t is the time step, j is the filter index and $ReLU\ (z) = max(0, z)$ is the ReLU activation function.

MaxPooling1D can be described as:

$$y_{t,j} = max\left(x_{t \cdot s+i,j}\right), \quad (2)$$

where x is the input from the previous layer, s is the stride (usually the same as the pool size), i ranges from 0 to the pool size minus 1 and j is the filter index.

The features extracted using CNN are transferred to the LSTM layer. Training is carried out through long-term state persistence and forward computation, and the backpropagation algorithm is used to train time series to create a forecasting model (Upadhya et al., 2021). We use 64 LSTM units to capture temporal dependencies, which can be mathematically described as follows:

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

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

$$\check{C}_t = tanh(W_C . [h_{t-1}, x_t] + b_C) \quad (5)$$

$$C_t = f_t . C_{t-1} + i_t . \check{C}_t \quad (6)$$

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

$$h_t = o_t . \tanh(C_t), \quad (8)$$

where f_t is the forget gate, x_t is the input at time t, i_t is the input gate, updating the cell state with new information and computed using the sigmoid activation function, $\check{C}_t$ is the new cell state, C_t is the cell state at time step t in the LSTM, o_t is the output gate, deciding the next hidden state h_t, h_t is the hidden state or the output of the LSTM at time t, σ is the sigmoid function, tanh is the hyperbolic tangent function, W_f , W_i , W_C , W_o are weight matrices, b_f, b_i, b_C, b_o are biases and "$\cdot$" denotes element-wise multiplication.

The flatten layer transforms the multidimensional output of the LSTM into a 1D vector to prepare it for the fully connected layers. For an input tensor of shape (n, m), the output will be a tensor of shape (n × m).

A fully connected layer with 128 units and ReLU activation is used for the classification of CN cybersecurity incidents. The input to this layer is the output data of the LSTM layer. For regularization, the Dropout technique is used at a rate of 0.5. A fully can be mathematically described as follows:

$$y = ReLU(W_x + b), \quad (9)$$

where W are the weights, x is the input, b are the biases and $ReLU\ (z) = max(0, z)$ is the ReLU activation function.

Dropout randomly deactivates some input units to 0 during each training update, thereby preventing overfitting. Its function can be described as follows:

$$y = \begin{cases} 0 & with\ probability\ p \\ \frac{x}{1-p} & with\ probablity\ 1-p \end{cases} \quad (10)$$

where p is the dropout rate (0.5 in this case).

The output layer contains several units corresponding to the classes in the target variable. It employs a softmax activation function to produce a probability distribution across the network traffic classes and can be described as follows::

$$y_i = \frac{exp(z_i)}{\sum_j exp(z_j)} \quad (11)$$

where $z = (W_x + b)$, W are the weights, b are the biases, i and j are indices over the output classes.

5. Dataset and experimens

We used the publicly available CICIDS2017. The CICIDS2017 dataset have 15 types of traffic, including BENIGN, DDoS, DoS Slowhttptest, Brute Force, Infiltration, DoS slowloris, XSS, Bot, DoS Hulk, Heartbleed, FTP-Patator, PortScan, DoS GoldenEye, SSH-Patator, and SQL Injection. The Heartbleed and Infiltration attacks were combined under the name Other. After analyzing, were selected: BENIGN, DDoS, DoS Slowhttptest, Brute Force, DoS slowloris, XSS, Bot, DoS Hulk, FTP-Patator, PortScan, DoS GoldenEye, SSH-Patator, SQL Injection, and Other (Table 1). The data was then cleaned by removing unnecessary columns, and rows containing NaN values were deleted.

Table 1. CICIDS2017 dataset traffic types

Traffic type	Statistics
BENIGN	2357511
DDoS	41834
DoS slowloris	5796
DoS Slowhttptest	5499
DoS GoldenEye	10293
DoS Hulk	230124
FTP-Patator	7935
PortScan	158804
Brute Force	1507
SSH-Patator	5897
XSS	652
Bot	1956
SQL Injection	21
Other	47

For class imbalance reduction, classes with a count above a specified threshold were identified for reduction. For each identified class, a random sample of instances equal to the threshold was taken and concatenated with instances from other classes to form the final reduced dataset. The class labels were one-hot encoded using the One Hot Encoder. A Random Forest Classifier (RFC) was trained on the dataset, and feature importances were extracted. The experimental setup involves feature selection using a RFC and CNN-LSTM model training. To allocate computational resources appropriately, 12 features were selected, which also led to a decrease in the training time of models.

As can be seen from Table 1, the network traffic classes are imbalanced. This can lead to unbalanced network traffic, allowing malicious cyberattacks to lurk in large volumes of normal data.

We employ data-level solutions to address the class imbalance problem. These techniques are categorized into undersampling, oversampling, and hybrid approaches (Buczak and Guven, 2016).

To address the imbalance in network traffic classes in the CICIDS2017 dataset, we applied both undersampling and oversampling techniques. Initially, to reduce class imbalance, we set a threshold to limit the maximum number of instances per class. This strategy ensures a balanced class distribution by capping the number of instances in classes that exceed a certain threshold. The threshold was set based on a preliminary analysis of the data distribution and the impact on model performance. We empirically set this threshold at 250,000 instances per class. Classes with more instances than this threshold were reduced to 250,000 instances, while those with fewer instances remained unchanged. This method is simple and serves as a solid baseline.

Then, we used the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) to address class imbalance. After balancing, the dataset was divided into training and test sets (70-30 split).

CNN-LSTM model were trained on the CICIDS2017 dataset, as depicted in Figure 3 and employ a Dense layer for the final output. CNN-LSTM model were developed and validated on the Google Colab platform using Python 3.10.12 with the CPU accelerator. The chosen DL framework was Keras, built on TensorFlow 2.14.0. The training involved 30 epochs with a batch size of 64.

Experiments have demonstrated that an increase in the number of epochs resulted in an improvement in loss and accuracy values for classification. Specifically, epochs of 30 and a batch size of 64 produced the best loss and accuracy values in classification. The learning process of the model was visualized using loss and accuracy curves (Figure 4). After training, CNN-LSTM model were tested, and their accuracy reached 0.9676.

The proposed approach to network cybersecurity incidents multi-classification based on DL was compared with other existing approaches to assess its effectiveness and applicability. Table 2 compares CNN-LSTM with various DL methods, including DNN (Deep Neural Network) (Vinayakumar et al., 2019), MLP (Toupas et al., 2019), MLP (Multi-Layer Perceptron), LSTM, and 1D-CNN (Roopak et al., 2019), DBN (Deep Belief Network) (Belarbi et al., 2022) and DeepGFL (Deep Graph Feature Learning) (Yao et al., 2018), using precision, recall and F1-score metrics.

Table 2. Performance evaluation of the proposed approach with state-of-the-art methods

References	Method	Metrics		
		Precision	Recall	F1-score
Vinayakumar et al. (2019)	DNN	0.908	0.973	0.939
Toupas et al. (2019)	MLP	0.943	0.956	0. 941
Roopak et al. (2019)	MLP	0.884	0.862	0.872
Roopak et al. (2019)	LSTM	0.984	0.898	0.895
Roopak et al. (2019)	1D-CNN	0.981	0.901	0.939
Belarbi et al. (2019)	DBN	0.887	0.997	0.940
Yao et al. (2018)	DeepGFL	0.948	0.448	0.531
Proposed approach	CNN--LSTM	**0.971**	**0.968**	**0,969**

Most studies in Table 2, such as those using DNN (Vinayakumar et al., 2019) and MLP, LSTM, and 1D-CNN (Roopak et al., 2019), focus on binary classification rather than multiclass classification. Among these, the DNN and 1D-CNN models proved to be the most successful, each achieving the highest F1-score of 0.939.

We compared our method against those shown in Table 2 and found that it achieves higher scores in terms of precision, recall, and F1-score, reaching 0,969. Additionally, our method is 3.1% and 3.2% more accurate in F1-score compared to MLP (Toupas et al., 2019) and (Belarbi et al., 2022), respectively, both of which also perform multiclass classification. It is worth noting that DeepDFL (Yao et al., 2018) conducts multiclass classification for 12 classes, achieving a high precision of 0.948 but a low recall of 0.448.

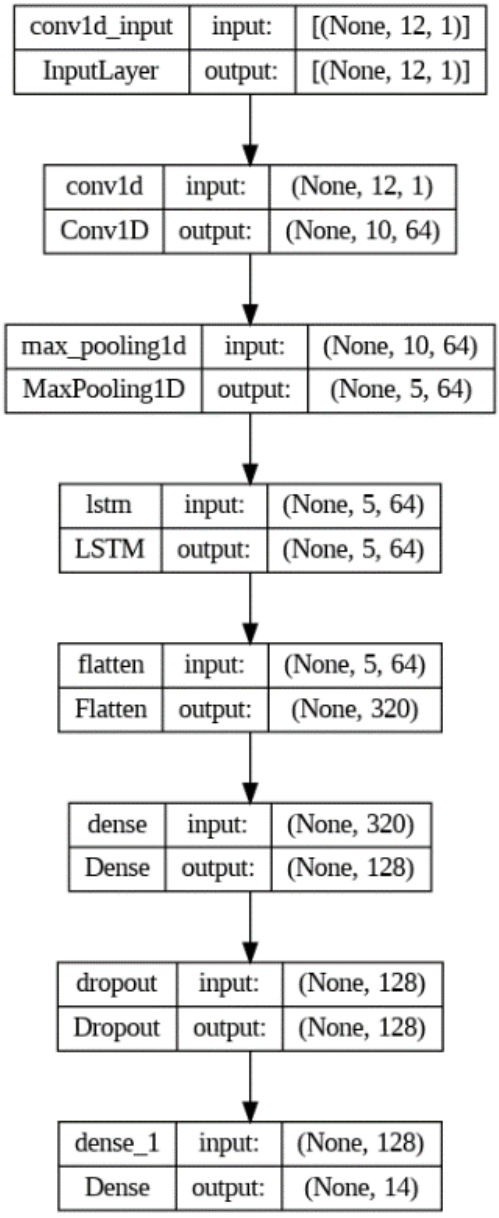

Fig. 3. CNN-LSTM model architecture.

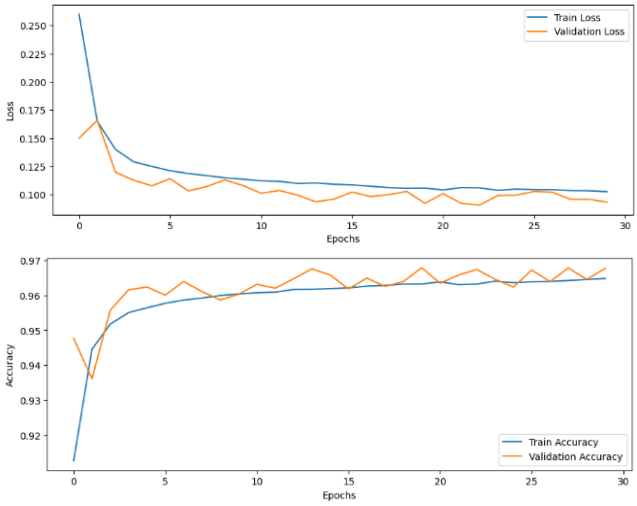

Fig. 4. CNN-LSTM model evaluation

Figure 5 show the confusion matrice depicting the performance of the CNN-LSTM model in classifying network traffic. Analysis of the confusion matrice reveals that, in classifying network traffic, the proposed model demonstrated superior accuracy.

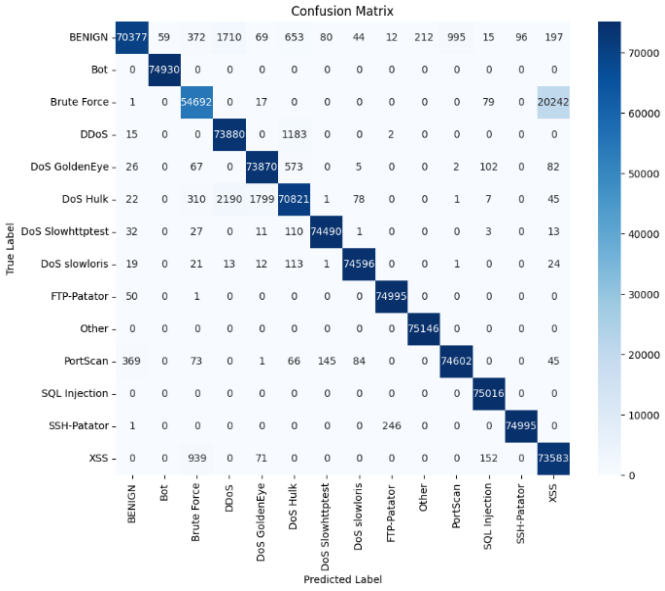

Fig. 5. CNN-LSTM model confusion matrix

Table 3 shows the assessment of the classification performance of network traffic. The experimental results indicate that the evaluation metrics for attack traffic classification of the CNN-LSTM model are generally higher than those of the CNN and LSTM models. Specifically, the CNN-LSTM model achieves 100% precision, recall, and F1 score across all evaluation criteria for, DoS Slowloris, PortScan, DoS Slowhttptest and SQL Injection traffic, in contrast to the CNN and LSTM models. Regarding precision, it also outperforms the CNN and LSTM models for DoS Hulk traffic.

Overall, the experimental results suggest that the CNN-LSTM model excels in classifying CN cybersecurity incidents compared to the CNN and LSTM models.

Table 3. Assessing the classification performance of network traffic

Traffic types	Model								
	CNN-LSTM			CNN			LSTM		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
BENIGN	0.99	0.94	0.97	**1.00**	0.97	0.98	0.99	0.93	0.96
DDoS	0.95	0.98	0.97	0.95	0.96	0.95	0.96	0.96	0.96
DoS slowloris	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.99	**1.00**	**1.00**	**1.00**
DoS Slowhttptest	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.99	**1.00**	0.98	0.99
DoS GoldenEye	0.97	0.99	0.98	0.94	0.99	0.96	0.99	0.94	0.96
DoS Hulk	0.96	0.94	0.95	0.93	0.90	0.92	0.89	0.96	0.93
FTP-Patator	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
PortScan	0.99	0.99	0.99	0.97	0.99	0.98	0.97	0.99	0.98
Brute Force	0.97	0.73	0.83	0.98	0.72	0.83	0.97	0.73	0.83
SSH-Patator	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
XSS	0.78	0.98	0.87	0.78	0.99	0.87	0.78	0.99	0.87
Bot	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
SQL Injection	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	0.99	**1.00**	**1.00**
Other	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

6. Conclusion and future work

This study proposes a deep learning-based approach for the multi-classification of network cybersecurity incidents using a hybrid CNN-LSTM model. By combining the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) with the temporal sequence learning power of Long Short-Term Memory (LSTM) networks, the proposed model effectively addresses the complexity and volume of modern network traffic. Trained and evaluated on the CICIDS2017 dataset, the model achieved a classification accuracy of 96.76%, demonstrating its potential to handle diverse and sophisticated cyber threats.

The results highlight the effectiveness of the hybrid architecture in capturing both spatial and temporal patterns in network data, enabling more accurate detection and classification of various attack types, such as DDoS, malware, and brute force attacks. Additionally, the model's high accuracy and low error rate showcase its suitability for real-time cybersecurity monitoring systems.

In the future, efforts will focus on exploring ways to further enhance the performance of deep learning-based cybersecurity incident multi-classification and its applicability in real-time network cybersecurity monitoring systems.

References

Alzahrani, M.Y., Bamhdi, A.M. (2022). Hybrid deep-learning model to detect botnet attacks over internet of things environments. Soft Comput 26, 7721–7735. https://doi.org/10.1007/s00500-022-06750-4.

Belarbi, O., Khan, A., Carnelli, P., and Spyridopoulos, T. (2022). An intrusion detection system based on deep belief networks. Proceedings of the International Conference on Science of Cyber Security, Shimane, Japan, 10–12 August 2022, Volume 13580, 377–392. https://doi.org/10.48550/arXiv.2207.02117.

Buczak, A. L., and Guven, E. (2016). A survey of data mining and machine learning methods for cybersecurity intrusion detection. IEEE Communications Surveys and Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Chawla, N. V., Bowyer, K. W., O., L., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.48550/arXiv.1106.1813

Chen Z. (2020). Deep Learning for Cybersecurity: A Review. International Conference on Computing and Data Science (CDS), 01-02 August 2020, Stanford, CA, USA. https://doi.org/10.1109/CDS49703.2020.00009.

Halbouni A., Gunawan T., S., Habaebi M., H., Halbouni M., Kartiwi M., Ahmad R. (2022). CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. IEEE Access, 10, 99837 – 99849. https://doi.org/10.1109/ACCESS.2022.3206425.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Juyal A., Bhushan B., Hameed A. A., Jamil A. (2023). Deep Learning Approaches for Cyber Threat Detection and Mitigation Proceedings of the 2023 7th International

Conference on Advances in Artificial Intelligence, 34-41. https://doi.org/10.1145/3633598.3633605.

Mbona I., and Eloff Jan H. P. (2022). Detecting Zero-Day Intrusion Attacks Using Semi-Supervised Machine Learning Approaches. IEEE Electron Device Letters 10(2): 69822-69838. http://dx.doi.org/10.1109/ACCESS.2022.3187116.

Medvedieva K., Tosi T., Barbierato E., Gatti A. (2024). Balancing the Scale: Data Augmentation Techniques for Improved Supervised Learning in Cyberattack Detection. Eng, 5(3), 2170-2205. https://doi.org/10.3390/eng5030114.

Otokwala, U., Petrovski, A., Kalutarage, H. (2021). Improving intrusion detection through training data augmentation. Proceedings of 14th International conference on Security of information and networks 2021 (SIN 2021), article 17. https://doi.org/10.1109/SIN54109.2021.9699293.

Roopak, M., Tian, G. Y., and Chambers, J. (2019). Deep learning models for cybersecurity in IoT networks. Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019, 452–457. https://doi.org/10.1109/CCWC.2019.8666588.

Toupas, P., Chamou, D., Giannoutakis, K. M., Drosou, A., and Tzovaras, D. (2019). An intrusion detection system for multi-class classification based on deep neural networks. Proceedings of the 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019, 1253–1258. https://doi.org/10.1109/ICMLA.2019.00206.

Tripathy, S., and Singh, R. (2021). Convolutional neural network: An overview and application in image classification. Proceedings of Third International Conference on Sustainable Computing, 145–153. https://doi.org/10.1007/978-981-16-4538-9_15.

Upadhya, K., Kaur, P., and Prasad, S. (2021). A review on data-level approaches to address the class imbalance problem. International Conference on Recent Challenges in Engineering Science and Technology (ICRCEST), 2K21, 152–158.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., and Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334.

Xiao, Y., Xing, C., Zhang, T., and Zhao, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access, 7, 42210–42219. https://doi.org/10.1109/ACCESS.2019.2904620

Yao, Y., Su, L., and Lu, Z. (2018). DeepGFL: Deep feature learning via graph for attack detection on flow-based network traffic. Proceedings of the MILCOM 2018 - IEEE Military Communications Conference (MILCOM), 579–584. . https://doi.org/10.1109/MILCOM.2018.8599821.

Zhang, Y., Xu, C., Jin, L., Wang, X. Guo, D. (2019). Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. IEEE Access, 7, 37004–37016. https://doi.org/10.1109/ACCESS.2019.2905041.