IRSTI 28.23.37 https://doi.org/10.26577/jpcsit2024-v2-i4-a1

A. Serdaly[1,*], D. Imabekkyzy[1], A. Akhmetay[1], B. Omarov [1,2]

[1]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[2]International Information Technology University, Almaty, Kazakhstan
*e-mail: altynayserdaly@gmail.com

IMPROVING NETWORK INTRUSION DETECTION USING THE MINI-VGGNET ARCHITECTURE: TACKLING CHALLENGES OF IMBALANCED DATA

Abstract. In the field of cybersecurity, the detection of network intrusions is a pressing challenge, particularly when dealing with imbalanced datasets. This study presents a novel model based on the MINI-VGGNet architecture, tailored specifically for identifying various types of network attacks using the CICIDS2017 dataset. The objective is to enhance detection accuracy while effectively managing the challenges posed by imbalanced data. The proposed model incorporates convolutional layers to capture deep features from network data, allowing for improved classification of 15 distinct classes of attacks, including DoS and DDoS. Experimental results demonstrate that the model achieves high accuracy in classifying common attack types, although challenges remain in accurately identifying specific classes like Web Attack – XSS and SQL Injection. The architecture's efficiency and lower computational demands make it suitable for real-world applications, particularly in resource-constrained environments. The findings indicate that further refinement of data balancing techniques is necessary to improve classification performance across all attack types. Overall, this research showcases the effectiveness of the MINI-VGGNet-Intrusion model in advancing intrusion detection systems and highlights the ongoing need for innovation in methods for handling imbalanced cybersecurity datasets.

Key words: deep learning, neural network, intrusion detection, imbalanced data, MINI-VGGNet.

1. Introduction

With the rapid development of information technology and the increasing volume of network traffic, cybersecurity has become one of the primary concerns for organizations worldwide. Network attacks can lead to severe consequences, including data breaches, financial losses, and disruption of critical systems. The efficient detection and prevention of network attacks is a priority for ensuring the security of information systems. Key methods to combat cybercrime include antivirus software, firewalls, and Intrusion Detection Systems (IDS) [1]. IDS is a tool designed to monitor network traffic and host activity to detect unauthorized or malicious actions. IDS can be divided into two types: Host-based IDS (HIDS), which monitors the activities on a specific device, and Network-based IDS (NIDS), which analyzes network traffic in real-time. To detect attacks, IDS relies on two main techniques: signature-based detection (identifying known attack patterns) and anomaly-based detection (spotting deviations from normal behavior). Major challenges for IDS include false positives and difficulties in detecting new attacks. Modern approaches incorporating machine learning and deep learning help improve the accuracy and adaptability of these systems, especially when dealing with imbalanced data [2].

One of the critical challenges in detecting network intrusions is data imbalance. In real-world scenarios, network traffic predominantly consists of normal data, while attack instances are much rarer. Modern approaches to network intrusion detection increasingly rely on deep learning techniques to address this issue. Deep neural networks, such as LSTMs and CNNs, are capable of identifying complex patterns in network data and effectively classifying both normal traffic and various types of attacks. However, even when using deep models, the problem of imbalanced data remains significant, requiring additional methods to ensure high detection accuracy [3]. In recent years, there has been growing interest in applying deep neural networks for network intrusion detection. Research has demonstrated that architectures such as Convolutional Neural Networks (CNN) and Long Short-Term

Memory (LSTM) networks produce improved results compared to traditional methods. Nonetheless, even state-of-the-art deep learning techniques face the challenge of data imbalance, necessitating further studies and the development of innovative approaches to enhance their performance in real-world scenarios.

Deep Learning and Neural Network Architecture. Deep learning is applied to artificial neural networks, which consist of artificial neurons organized into layers. In such networks, input data passes from the first layer, known as the input layer, through intermediate (hidden) layers, to the final (output) layer, where the data is processed. If there are multiple intermediate layers, the network is referred to as "deep." Neurons are associated with a function called an activation function. The activation function's value depends on the weighted sum of the neuron's inputs and a threshold value. The output of a neuron is the result of applying the activation function to the scalar product of the input vector and the vector of the neuron's weights, modified by a given bias [4]. Examples of nonlinear functions used as activation functions include the sigmoid, softmax function, rectified linear unit (ReLU), and hyperbolic tangent.

Training a neural network involves using a loss (or error) function, which characterizes the difference between the true value of the target variable and the value predicted by the network. The configuration of neurons, layers, and the connections between them gives rise to different neural network architectures. Thus, deep learning methods can be categorized based on the architecture of the neural networks in which they are applied. Below is a brief description of the main deep learning techniques encountered in the analysis of related works.

Artificial Neural Networks (ANN), as mentioned earlier, can be classified as deep (Deep Neural Network, DNN) or shallow (Shallow Neural Network, s-NN) depending on the number of hidden layers. A basic form of these networks is the feed-forward neural network, where signals flow strictly from the input layer to the output layer. Training is typically carried out using the backpropagation method. The Multilayer Perceptron (MLP), introduced by F. Rosenblatt, is a simple form of a neural network based on a mathematical model that mimics how the brain processes information [5]. MLP is a fully connected ANN that may have one or more hidden layers between the input and output layers [6].

A Convolutional Neural Network (CNN) is a multi-layered, feed-forward network comprising alternating convolutional layers and subsampling (or pooling) layers. The convolutional layer generates a feature map by performing element-wise multiplication of a weight matrix (convolution kernel) on each fragment of the input layer and then adding the result to the corresponding position in the output layer. CNNs were originally used for image processing but are now applied to various other tasks. In terms of network intrusion detection, applying CNN requires transforming and normalizing each feature vector in the dataset into a grid-like structure or a conditional "image" [7].

A Recurrent Neural Network (RNN) forms a directed sequence of connections between neurons that possess internal memory and can feed data back into the network. This architecture allows RNNs to exhibit dynamic behavior over time and enables them to process sequential data of arbitrary length. RNNs can be trained with varying degrees of supervision, with the LSTM and GRU variants being the most common. Long Short-Term Memory (LSTM) networks were developed to address the vanishing gradient problem inherent to RNNs. In an LSTM, memory retention is handled by specialized structures called "gates," which regulate the flow of information. The LSTM module allows values to be stored both short-term and long-term.

A Gated Recurrent Unit (GRU) uses a similar structure to the LSTM but with fewer gates, resulting in fewer parameters and reduced computational resources required for training. Bidirectional GRU (BGRU) and Bidirectional LSTM (BiLSTM) are variants of GRU and LSTM, respectively. They allow networks to predict outcomes not only based on processed data but also by considering the entire sequence. In these networks, computation occurs in two directions, with the output layer's neurons considering both past and future context to generate predictions [8].

2. Literature Review

This section provides an analysis and comparison of relevant studies and the deep learning methods described within them. Most of the studies analyzed in this work were selected from the Google Scholar database [9] using the "Publish or Perish" tool [10].

In [11], the authors applied a CNN-BiLSTM approach to solve the problem of network intrusion detection, where CNN captures spatial features and BiLSTM handles temporal features. A hybrid sampling technique (OSS combined with SMOTE)

was used to reduce training time and balance the dataset. Following the hybrid sampling, the training time decreased across all compared models. While CNN-BiLSTM showed a lower training speed than LeNet-5, it performed better in all other metrics. The comparison was conducted using the NSL-KDD and UNSW-NB15 datasets.

In [12], a deep neural network (DNN) with three hidden layers outperformed classic machine learning algorithms for network intrusion detection. The comparison was performed on the KDD Cup 99 dataset using DNNs with 1-5 hidden layers, alongside algorithms such as Ada Boost, Decision Tree, K-Nearest Neighbor, Linear Regression, Naive Bayes, Random Forest, SVM*-Linear, and SVM*-rbf. However, the authors emphasized the need for further studies on modern datasets and real-world environments, particularly in adversarial conditions.

Another study [13] developed a hybrid IDS based on a convolutional recurrent neural network for network intrusion detection, where CNN captures spatial features, and RNN identifies temporal patterns. To address data imbalance, oversampling of minority class instances was used. To enhance generalization capabilities and minimize neuron retraining, Gaussian Noise layers were added before the CNN and RNN layers. The model's performance was evaluated using the CSE-CIC-IDS2018 dataset, with comparisons made to methods in other studies and to models such as Decision Tree, Logistic Regression, and XGBoost. Despite the promising results, it is important to note that the evaluation sets in the experiments differed, and the reported scores do not match across comparisons.

In [14], CNN was applied to address the issue of network intrusion detection by initially transforming the data from a vector format into an "image" (matrix). Principal Component Analysis (PCA) and an Autoencoder (AE) were used to reduce the dimensionality of the feature space, while batch normalization (BN) was employed to optimize the training process. Compared to traditional machine learning algorithms (Naive Bayes, Logistic Regression, Decision Tree, Random Forest, SVM, Adaboost), RNN, and a three-layer DNN, the proposed model significantly reduced detection time and achieved good results on the KDD Cup 99 dataset. However, the model showed poor detection rates for the User to Root (U2R) and Remote to Local (R2L) attack types due to the limited number of examples for these attacks in the dataset–20.61% and 18.96%, respectively. The authors plan to address this issue in future research by generating attack examples using a Generative Adversarial Network (GAN).

In [15], a method for transforming data from the NSL-KDD dataset into binary vectors was proposed, from which "images" (matrices) were created for classification using CNN, bypassing the need for feature selection. CNN networks (ResNet 50 and GoogLeNet), tested on subsets of the transformed NSL-KDD dataset, outperformed standard classifiers but lagged behind state-of-the-art solutions. The comparison was made with classifiers such as j48, Naive Bayes, NB Tree, Random Forest, Random Tree, Multi-layer Perceptron, and SVM.

In [16], the use of Recurrent Neural Networks (RNN) for solving the problem of network intrusion detection was proposed. For binary and multi-class classification, RNN outperformed traditional algorithms (J48, ANN, RF, SVM, etc.) on the NSL-KDD dataset, though it required more training time. The proposed method also showed better results than a reduced RNN in the KDD CUP 1999 dataset [17]. The paper also discusses hyperparameter selection, highlighting how the number of neurons and learning rate affect model accuracy. The authors noted the problems of vanishing and exploding gradients and suggested further exploration of LSTM and Bi-directional RNN to resolve these issues.

In [18], a network intrusion detection model combining Artificial Neural Networks with Correlation-based Feature Selection (CFS) was proposed. The model was implemented using RapidMiner and tested on the NSL-KDD and UNSW-NB15 datasets. The use of CFS improved model accuracy, specificity, and sensitivity by reducing the data size and shortened computation time. The implemented model was compared with other modern approaches and demonstrated competitive results, though it required more computational resources. This approach could be used in various communication networks, including to protect Internet of Things (IoT) servers.

In [19], a model consisting of two neural networks was proposed to solve the problem of network intrusion detection: Shallow Neural Network (S-NN) and Deep-Optimized Neural Network (DONN). The S-NN is simple and fast, while the D-ONN is complex and slower. Feature selection was carried out using correlation analysis and an entropic approach. This model demonstrated good results on the KDD Cup 99 dataset. The authors also pointed out the potential use of this method for protecting wireless networks and IoT systems.

In the following paper [20], the BGRU+MLP model was proposed to address the network intru-

sion detection problem. Experiments were conducted using the KDD Cup 99 and NSL-KDD datasets. The results showed that GRU performs better than LSTM, BGRU surpasses GRU, and the combination of BGRU and MLP provides better results than using RNN (GRU or LSTM) or MLP individually. The BGRU+MLP model showed superior performance in terms of accuracy, the number of detected events, and the rate of false positives (FPR), but faced challenges in detecting R2L and U2R attack types. The authors noted that this issue is common across other systems due to the limited number of such attack examples in the dataset.

In [21], the SFSDT+RNN model was proposed to improve the detection accuracy of various attack types, including the challenging R2L and U2R attacks. The SFSDT hybrid algorithm was used for feature selection: Sequential Forward Selection (SFS) identified the most significant set of features, and a Decision Tree (DT) was employed to refine the feature set. Experiments were conducted on the NSL-KDD and ISCX 2012 datasets. The model using LSTM demonstrated the highest accuracy among the three RNN types (RNN, LSTM, GRU). The use of SFSDT for feature selection also reduced computation time and memory usage. However, it is important to note that the study did not provide detailed implementations of the RNN, LSTM, or GRU architectures used in the experiments. Special attention was given to studies providing analytical reviews on the application of deep learning in IDS.

In [22], a literature review was provided on the use of neural networks in IDS from 2015 to 2019. The review covered research papers, new methodological proposals, and educational articles. The most frequently used datasets in attack detection systems employing neural network architectures were discussed, along with both general and specific datasets and their utilization features. The issue of security implications when using neural networks in IDS was also raised.

In the following study [23], the authors presented a literature review on the use of machine learning and neural networks in IDS from the perspective of an IDS taxonomy they proposed. The review included commonly used machine learning algorithms, performance metrics, and IDS classifications based on datasets. Problems within the field and directions for future research were highlighted.

Research [24] provided an analytical review of deep learning methods for cybersecurity tasks. Various deep learning techniques were discussed in the context of specific cybersecurity applications. The authors concluded by discussing the applicability and unique characteristics of deep learning techniques for addressing cybersecurity challenges.

About half of the studies employ neural network methods. These methods are used for feature engineering (to partition or reduce the signature space), optimizing the learning process (e.g., data balancing or reducing retraining), and classification. Various types of RNNs and CNNs are frequently encountered. The combination of different architectures with a single method aims to eliminate specific method limitations or improve the level of automation in the overall process of detecting attacks. The predominance of works focusing on feature engineering methods, data processing, or various auxiliary methods (such as optimization techniques) underscores the significance of these steps in achieving high results with neural networks.

In the reviewed studies, the comparison of the proposed deep learning methods was primarily conducted against other deep learning approaches (including variations of the proposed method) or various machine learning techniques. Comparisons were often made using the KDD Cup 1999, NSL-KDD 2009, UNSW-NB15, and ISCX 2012 datasets. The most modern dataset analyzed was CSE-CIC-IDS2018 [25]. It is important to highlight a common issue with deep learning methods, characteristic of classical machine learning techniques: the use of outdated, imbalanced datasets that do not align with current data for training purposes. In the studies reviewed, the application of deep learning methods for detecting network intrusions often corresponds to a multi-class classification task. For quality assessment, all studies utilized metrics such as accuracy (ACC), precision, recall, F1-score, and detection rate (DR), with the proportion of correct responses being a common evaluation criterion in nearly half of the studies.

3. Research Methodology

The structure of the proposed model, shown in Figure 1, consists of three main modules: the imbalance handling module, the dimensionality reduction module, and the classification module. Each module has been optimized through hyperparameter search, based on experimentation and numerous trials, ensuring better performance outcomes.

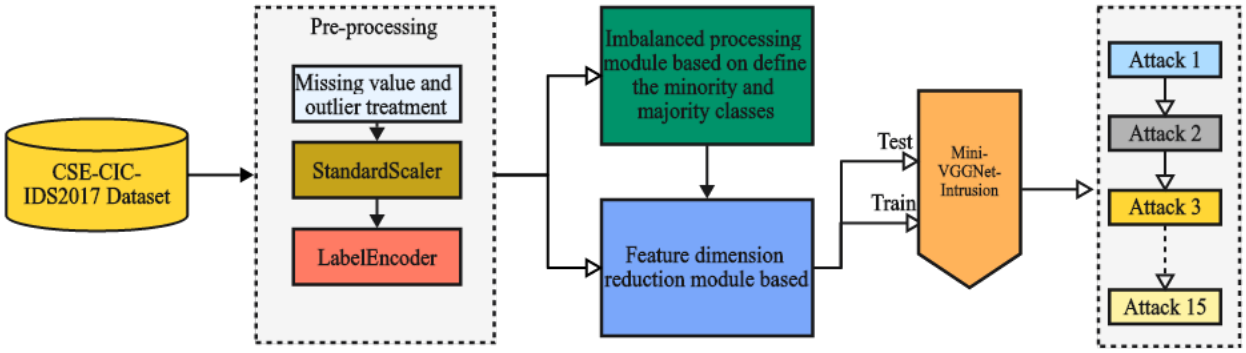

Figure 1 – Structure of the Model

To address the challenge of detecting network attacks using imbalanced data, a model based on the MINI-VGGNet architecture was developed utilizing the CICIDS2017 dataset. The CICIDS2017 dataset is a widely used resource for detecting network intrusions, created by the Canadian Institute for Cybersecurity in 2017. This dataset simulates real-world network traffic and encompasses a broad spectrum of attacks, including DoS, DDoS, PortScan, Brute Force, XSS, SQL Injection, and others. It includes various types of traffic, along with benign traffic labeled as "BENIGN," enabling the analysis of network anomalies apart from the attacks. This dataset is instrumental in training artificial intelligence and machine learning algorithms for efficient network traffic analysis and attack detection. Comprehensive preprocessing and selection procedures were developed for the dataset in question.

The primary advantage of this model lies in its ability to extract deep features through convolutional layers (Conv2D), making it well-suited for complex classification tasks, such as detecting different types of network attacks. The Mini-VGGNet-Intrusion model is designed to address the multiclass classification problem, covering 15 classes of network events ranging from benign traffic to various attacks like DoS, DDoS, PortScan, and others. The Mini-VGGNet-Intrusion is a specialized modification of the classic VGGNet architecture for network intrusion detection tasks. Originally proposed by Simonyan and Zisserman in 2014, VGGNet is renowned for its deep and powerful architecture tailored for image classification tasks. However, its use requires substantial computational resources due to its many layers and parameters. To address this, a simplified version, Mini-VGGNet, was developed, which retains the core principles of VGGNet while reducing the number of layers and filters. This version decreases the model's depth while maintaining high efficiency for processing smaller datasets, thereby requiring fewer resources.

The Mini-VGGNet-Intrusion model was trained using the CSE-CIC-IDS2017 dataset, a comprehensive source for network intrusion detection. Preprocessing and data handling involved several critical steps to ensure effective model training, including data standardization, imbalance handling, and dimensionality reduction. These processes aim to prepare clean, balanced, and optimized data for classification tasks.

When working with unbalanced data sets in classification problems, there are significantly more samples in one class (majority class) than in another class (minority class). This imbalance can lead to a model that works well in a majority class but does poorly in a minority class. Solving this problem often requires defining these classes and using methods for balancing data sets. Here is given for processing unbalanced data sets by separating minority and majority classes and defining "complex" patterns using the algorithm of near neighbors. This approach is most effective when working with data sets in which certain classes are not represented enough.

The module performs the following steps:

1. Identify minority and majority classes: determine which classes are minority and which are majority classes based on the analysis of class distribution.

2. Divide data into minority and majority classes: divide the data set into two subsets based on class labels.

3. Using the near neighbors method to identify complex patterns : applying the near neighbors pattern to the majority class to identify minority patterns that are difficult to classify.

4. Separation of complex and simple samples: using the sample of close neighbors to find samples in the minority class closest to the majority, which indicates that it can be difficult to classify them.

3.1. StandardScaler and Preprocessing.

StandardScaler was applied as a preprocessing step to normalize the input features. It ensured consistent scaling across feature ranges, a vital step before implementing machine learning models. In this dataset, the terms “minority class” and “majority class” refer to the distribution of samples across different categories in a classification problem:

- Majority Class: This is the category with the most samples in the dataset. For example, in this case, the “BENIGN” class represents the majority because it occurs far more frequently than others.

- Minority Class: This refers to categories with significantly fewer samples compared to the majority class. These classes are underrepresented in the dataset.

The imbalance between these classes creates a significant challenge for machine learning models. Since most samples belong to the majority class, models can achieve high overall accuracy by predominantly predicting the majority class. However, this approach undermines the performance on minority classes, making it difficult to detect less common but equally important events.

This issue is particularly critical in fields like fraud detection and spam filtering, where failing to identify minority classes correctly can lead to severe consequences. Achieving a balance between the accurate prediction of both majority and minority classes is essential for creating reliable and effective models.

3.2. Key Modules and Workflow.

The workflow to process the dataset and train the model is divided into four primary stages:

1. Data Preprocessing Module:

- Cleaning: Missing and outlier values were handled, with rows containing None, NaN, inf, or nan values removed.

- Normalization: The StandardScaler transformed data to a [0, 1] range for uniformity.

- Encoding: One-hot encoding was applied to discrete objects to convert categorical values into binary vector representations.

The processed data from this module was then forwarded to the imbalance handling module.

2. Imbalance Handling Module:

- Leveraging a combination of random undersampling and algorithms to divide data into majority and minority classes, this module ensured a balanced representation in the training dataset.

3. Dimensionality Reduction Module:

- High-dimensional data often contains redundant information. Dimensionality reduction techniques removed this redundancy, improving the reliability of the data while preserving critical features. The refined data was subsequently passed to the classification module for detecting multi-class anomaly attacks.

This integrated process enables effective model training by addressing data quality, class imbalance, and feature optimization, contributing to enhanced detection of various network intrusions.

The Mini-VGGNet-Intrusion model developed for this task represents an optimized adaptation of the original architecture for analyzing network data. Unlike the original Mini-VGGNet, which utilizes filters for image processing, this model is designed for one-dimensional or two-dimensional network data using 2×12 \times 12×1 filters. This approach effectively captures spatial and temporal features from the inputs. The model consists of two convolutional layers followed by MaxPooling2D, and it includes two fully connected layers for the final classification of 15 different classes of network attacks (see Figure 2). This architecture is specifically designed to address the classification of network attacks such as DoS, DDoS, and others, making it relatively easy to implement in efficient and resource-constrained real-world scenarios.

This figure 2 illustrates the architecture of the Mini-VGGNet-Intrusion model, represented using the visualkeras library. The model is designed using several convolutional layers to capture the spatial features of the data for classification tasks. It employs pooling to reduce the dimensionality of the data and prevent retraining. Fully connected layers following the convolution and pooling layers assist in classifying network intrusions by converting features into class probabilities. This model is used to classify network attacks into 15 different classes. It utilizes the Adam optimizer and sparse categorical crossentropy loss function for multi-class classification.

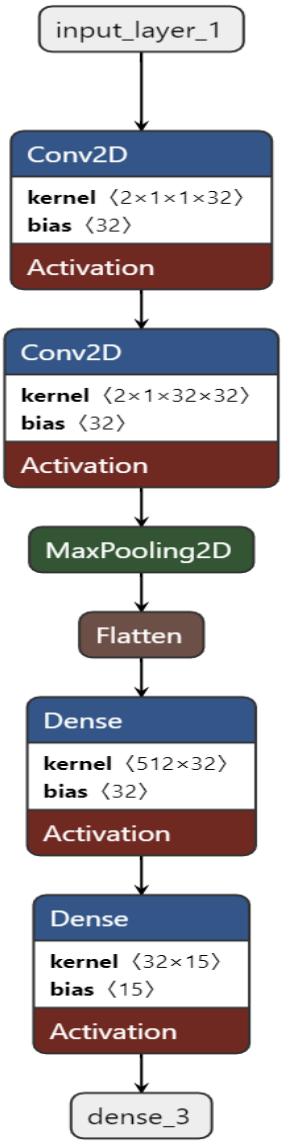

Figure 2 – Model Architecture

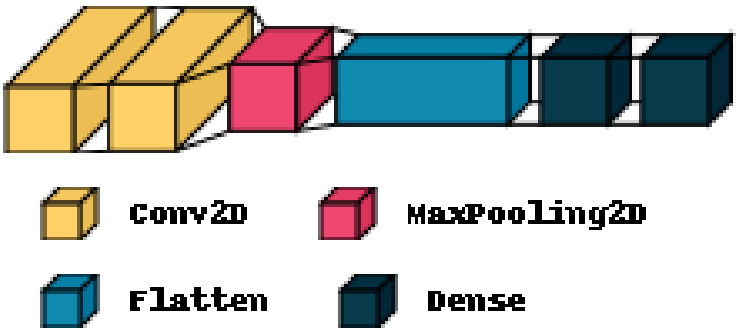

Figure 3 – Main Layers of the Model

Figure 3 shows a visualization of the Mini-VGGNet-Intrusion architecture, which illustrates the sequence of the model's main layers. The architecture begins with two convolutional layers (Conv2D) that extract key features from the data, followed by a MaxPooling2D layer that reduces the dimensions of the features and enhances computational efficiency. Next, the model includes a Flatten layer that transforms the two-dimensional data into a one-dimensional array for the dense layers. These layers are responsible for classifying the inputs into 15 classes. Figure 4 provides a description of the layers of the synthesized neural network model.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 18, 2, 32)	96
conv2d_1 (Conv2D)	(None, 17, 2, 32)	2,080
max_pooling2d (MaxPooling2D)	(None, 8, 2, 32)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 32)	16,416
dense_1 (Dense)	(None, 15)	495

Total params: 57,263 (223.69 KB)
Trainable params: 19,087 (74.56 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 38,176 (149.13 KB)

Figure 4 – Brief Description of the Model

3.3. Comparison with Modern Neural Network Models: Mini -VGGNet and CNN-LSTM

Among modern neural network architectures, Mini-VGGNet is an efficient and simple model that can achieve high results in many cases. However, recent research often utilizes hybrid models like CNN-LSTM, which combine multiple architectures to process both spatial and temporal dependencies effectively. Comparing Mini-VGGNet with CNN-LSTM highlights the advantages and limitations of the proposed model in the context of modern approaches.

Comparison of Mini -VGGNet and CNN-LSTM

1. Architecture:

- Mini-VGGNet is primarily designed to process spatial data. Its core consists of convolutional layers (Conv2D), which are optimal for extracting spatial features from input data. This model achieves high efficiency by reducing the number of layers.

- CNN-LSTM is a hybrid architecture, where the Convolutional Neural Network (CNN) extracts spatial features, and the Long Short-Term Memory (LSTM) layer processes temporal dependencies. This architecture combines the strengths of CNN for spatial data and LSTM for sequential data, making it suitable for handling both spatial and temporal information.

2. Data Processing:

- Mini -VGGNet does not account for temporal dependencies, which limits its ability to process time-series or sequential data effectively. While it performs well with spatial data, its ability to handle time-dependent patterns, such as those found in network traffic or intrusion attempts, is limited.

- CNN-LSTM combines CNN's ability to process spatial data with LSTM's strength in handling sequential data. This makes the CNN-LSTM model more suitable for tasks involving time-series data, where temporal dependencies are crucial, such as in detecting DDoS attacks or identifying network intrusion patterns over time.

3. Efficiency:

- Mini -VGGNet is highly effective at capturing spatial features but does not perform well when temporal patterns are involved. As a result, this model can be less accurate in identifying attacks

that have temporal characteristics, such as DDoS attacks, which unfold over time.

- CNN-LSTM is highly efficient in processing both spatial and temporal data. It integrates CNN's feature extraction capabilities with LSTM's long-term dependency modeling, enabling it to detect complex attack patterns that may unfold over time. This model shows significant improvement in performance when handling dynamic attack patterns.

4. Application Areas:

- Mini -VGGNet is typically effective in static or low temporal-dependency data, such as identifying port scanning activities or static network intrusions. It works well when the focus is primarily on spatial features.

- CNN-LSTM, on the other hand, excels in scenarios that involve sequential or time-varying data, such as detecting DDoS attacks, SQL injections, or any other attack with a dynamic temporal pattern. It is better equipped to capture complex attack patterns that evolve over time.

5. Challenges and Limitations:

- Mini-VGGNet has a simplified architecture, which limits its ability to process temporal dependencies. As a result, it may not perform as well in detecting advanced, time-based attacks, which rely on identifying changes in network behavior over time.

- CNN-LSTM often requires more computational resources and longer training times due to the integration of both CNN and LSTM components. However, this model provides better accuracy in tasks that require analyzing time-series or sequential data, such as detecting complex attack behaviors or anomalies in network traffic.

Mini-VGGNet performs excellently with spatial data but lacks the ability to model temporal dependencies effectively, which limits its performance when detecting dynamic and evolving attacks. On the other hand, CNN-LSTM overcomes these limitations by combining the spatial feature extraction power of CNN with the temporal modeling capabilities of LSTM, making it more effective in identifying complex, time-dependent attack patterns. While the Mini-VGGNet model is efficient for static intrusion detection tasks, CNN-LSTM provides a more robust solution when both spatial and temporal information is necessary for accurate prediction, such as in the case of DDoS attacks or SQL injections. Therefore, for dynamic and time-sensitive tasks, hybrid models like CNN-LSTM are preferred over simpler architectures like Mini-VGGNet.

The LSTM model achieves an accuracy of 0.989, demonstrating its strong ability to handle sequential data and capture temporal dependencies in network traffic, making it highly effective for intrusion detection over time. In comparison, the Mini-VGGNet-Intrusion model reaches an accuracy of 0.985, showing excellent performance in classifying network intrusions by extracting spatial features through convolutional layers. While both models perform well, the LSTM slightly outperforms the Mini-VGGNet-Intrusion model, especially in tasks requiring temporal data processing, whereas Mini-VGGNet-Intrusion excels in spatial feature extraction.

4. Research Results and Discussions

This section focuses on the experimental results obtained using the proposed model's architecture (Figure 5).

The graph displays ROC curves for 15 classes, illustrating the relationship between true positive and false positive rates across various classification thresholds (Figure 6). The ROC curves indicate that the model effectively classifies the majority of classes, such as BENIGN, DoS Hulk, and DDoS, where the AUC value is 1.00. However, for the Web Attack – XSS and Web Attack – SQL Injection classes, the AUC values are 0.58 and 0.81, respectively, highlighting the need to improve the model's accuracy. The AUC value for the Infiltration class was not computed, potentially due to insufficient data or the model's inability to accurately identify this class. The gray dotted line in the graph represents random classification with an AUC of 0.5; the farther the ROC curve is from this line, the better the model's classification performance.

The performance and training efficiency of the model were assessed using the Loss Function and Model Accuracy graphs (Figure 7). Loss Function Graph: This graph illustrates the model's error. A decreasing loss value indicates that the model is learning, while it also helps identify overfitting. Accuracy Graph: This graph displays the percentage of correct predictions made by the model. By comparing training and validation accuracy, it is possible to detect overfitting and monitor the model's performance. These graphs provide insights into the model's effectiveness and allow for adjustments to its parameters when necessary.

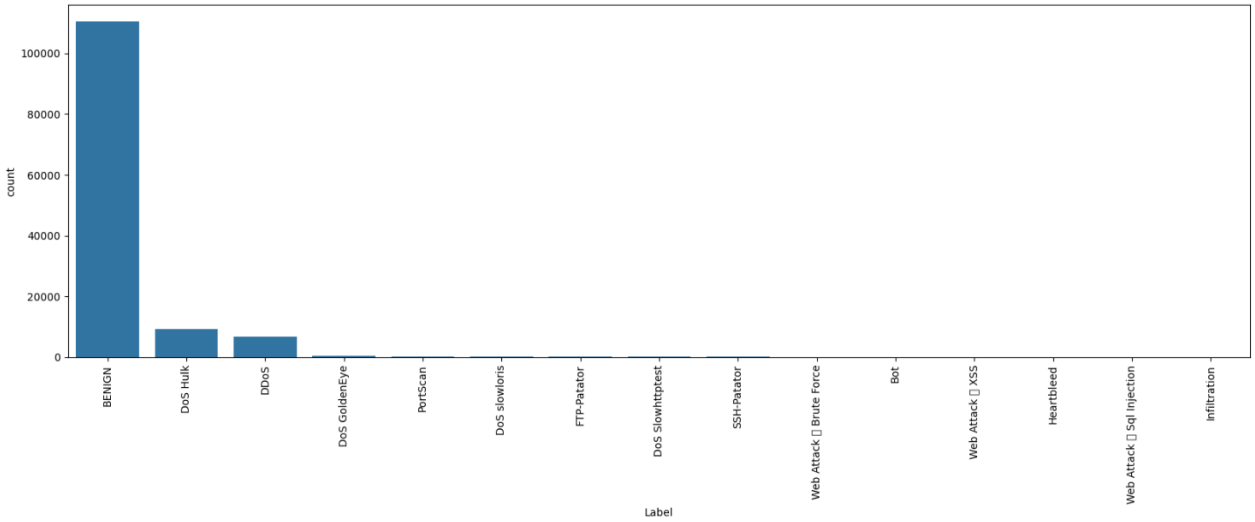

Figure 5 – Comparison by Distribution of Attacks

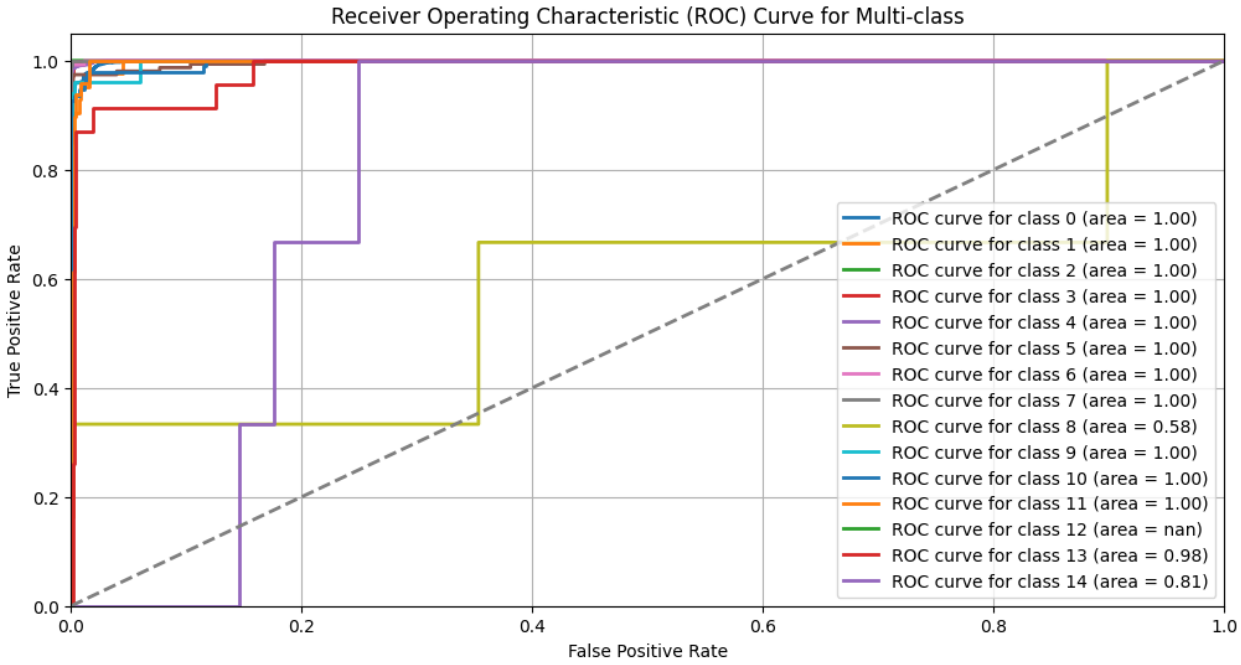

Figure 6 – AUC-ROC Curve for Intrusion Detection

This figure 8 presents the confusion matrix, which allows for the analysis of errors, calculation of quality metrics for the model, and identification of class imbalance. The Mini-VGGNet-Intrusion model has advantages over traditional machine learning models due to its ability to automatically extract complex features from data for intrusion detection. This capability improves classification accuracy and reduces the need for manual data processing. A table of comparison of the results obtained in Table 1 shows the effectiveness of the proposed deep learning and machine learning models.

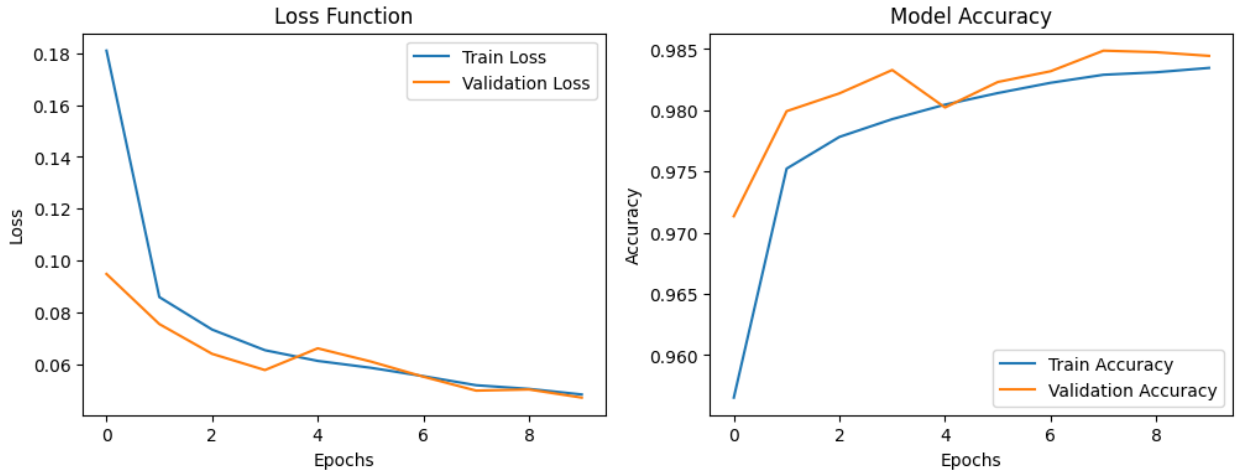

Figure 7 – Loss Function and Model Accuracy

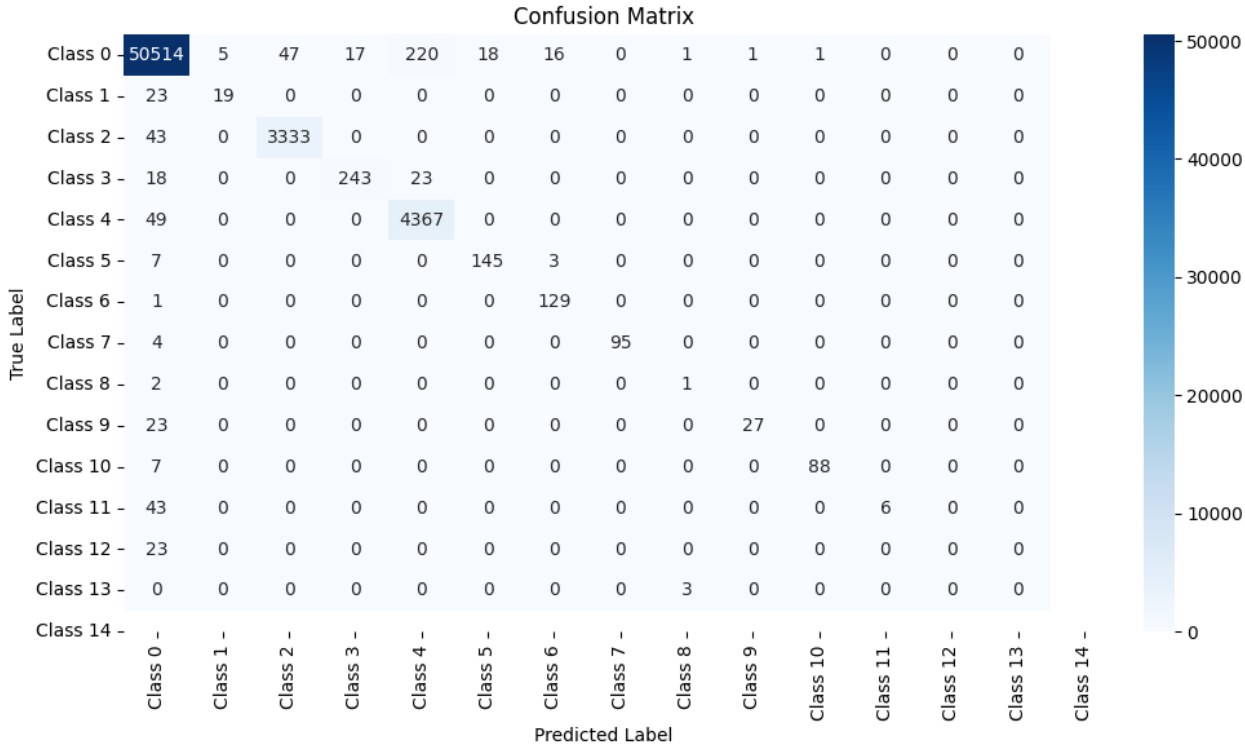

Figure 8 – Model's Confusion Matrix

Table 1 – Comparison of Obtained Results

Methods Used	Models	Accuracy	Precision	Recall	F-score	ROC
Proposed Deep Learning Model	Mini-VGGNet-Intrusion	0.985	0.988	0.993	0.987	0.98
Machine Learning Models	NB	0.874	0.832	0.863	0.851	0.92
	RF	0.851	0.854	0.822	0.856	0.77
	DT	0.602	0.524	0.585	0.642	0.65
	SVM	0.873	0.852	0.862	0.851	0.78
	KNN	0.856	0.839	0.831	0.837	0.92
	LR	0.862	0.853	0.837	0.858	0.78

The proposed model is a modification of the well-known VGGNet architecture, tailored for network intrusion detection tasks. However, its complexity and resource intensity require significant computational resources, making it less suitable for tasks with limited resources or smaller data volumes. To simplify the architecture, MINI-VGGNet was introduced, retaining the core principles of the original model while reducing the number of layers and adapting it for analyzing network data. The Mini-VGGNet-Intrusion model differs from MINI-VGGNet by using specialized dimensional filters designed to effectively extract spatial and temporal features from network data, rather than standard filters intended for image processing. The model consists of two convolutional layers, one MaxPooling2D layer, and two fully connected layers. This architecture is specifically designed for accurately detecting 15 different classes of network attacks, including DoS and DDoS. The model's straightforward structure and low resource requirements allow it to conserve computational resources while maintaining high classification accuracy compared to deeper networks. Despite its many advantages in detecting unauthorized network intrusions, the proposed model also has some limitations. Firstly, although it efficiently captures temporal sequences, its complexity and computations can require significant time and resources, potentially posing challenges when speed is critical for network security. Secondly, it may not always be suitable for tasks with limited resources or smaller data volumes.

5. Conclusion

In summary, the proposed Mini-VGGNet-Intrusion model has demonstrated significant achievements and results in detecting network intrusions, particularly in the context of imbalanced data. Its architecture, based on convolutional layers, provides high classification accuracy for various types of attacks, making it suitable for use in intrusion detection and prevention systems (IDS/IPS). However, additional configuration and application of methods to address data imbalance are necessary to improve the classification of certain attacks. The Mini-VGGNet-Intrusion effectively adapts the classic VGGNet architecture for network intrusion detection tasks. By simplifying the architecture and optimizing for network data, the model efficiently extracts features while maintaining low computational costs, ensuring accurate classification. This makes it a viable option for situations where traditional deep neural networks may demand excessive resources. Thus, Mini-VGGNet-Intrusion presents an effective solution for network intrusion detection, striking a balance between accuracy and computational resource requirements.

To enhance the conclusion with a focus on the MINI-VGGNet model, you can expand the future research directions as follows:

Future research in the area of network intrusion detection with the MINI-VGGNet model can explore several promising directions. One key focus could be optimizing the current architecture to improve its efficiency and accuracy, particularly in detecting rare and complex attack types. Hybrid models, combining the strengths of CNNs like MINI-VGGNet with other techniques, such as LSTMs, could further enhance the model's ability to capture both spatial and temporal features of network traffic. Additionally, addressing the issue of class imbalance is crucial for improving performance on minority classes. Advanced data

balancing techniques, such as the use of Generative Adversarial Networks (GANs) for generating synthetic data or other oversampling methods, could be explored to improve the model's detection capabilities for less common attack types. Furthermore, adapting MINI-VGGNet for real-time intrusion detection is an important research direction. This would require optimizing the model for faster processing times without sacrificing accuracy, enabling real-time threat detection in dynamic network environments. Finally, expanding the model's application by incorporating new, more diverse datasets and real-time threat intelligence could help the model stay adaptable to emerging cyber threats. These research directions will contribute to the ongoing development of more accurate, efficient, and scalable intrusion detection systems based on MINI-VGGNet.

Author Contributions

Conceptualization, B.S.; Methodology, D.I.; Software, A.A. and A.S.; Validation, D.I.; Formal Analysis, A.S. and B.S.; Investigation, A.S., D.I., A.A., B.S. ; Resources, A.S.; Writing –Original Draft Preparation, A.S., D.I., A.A.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. D. I. Sychev, "Machine and Deep Learning Methods for Intrusion Detection Systems: Review and Analysis," *International Journal of Information Technologies and Energy Efficiency*, vol. 8, no. 4(30), pp. 9–17, 2023.
2. H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: a comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013.
3. S. Mohammadi and A. Namadchian, "Anomaly-based Web Attack Detection: The Application of Deep Neural Network Seq2Seq With Attention Mechanism," *The ISC International Journal of Information Security*, vol. 12, no. 1, pp. 44–54, 2020, DOI: 10.22042/isecure.2020.199009.479.
4. D. A. Gayfulina and I. V. Kotenko, "Application of Deep Learning Methods in Cybersecurity Tasks. Part 1,» *Cybersecurity Issues*, no. 3 (37), pp. 76–86, 2020, DOI: 10.21681/2311-3456-2020-03-76-86.
5. F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, DOI: 10.1037/H0042519.
6. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, Eds., MIT Press, 1985, pp. 318–362.
7. I.Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
8. E. Culurciello, "The fall of RNN / LSTM," *Towards Data Science*, 2018. [Online]. Available: https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0
9. Google Scholar. [Online]. Available: https://scholar.google.com [Accessed: Oct. 4, 2023].
10. A.W.Harzing, *Publish or Perish*, 2007. [Online]. Available: https://harzing.com/resources/publish-or-perish
11. K.Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020, DOI: 10.1109/ACCESS.2020.2973730.
12. R. K. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1–6, DOI: 10.1109/ICCCNT.2018.8494096.
13. M. A. Khan, "HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System," *Processes*, vol. 9, no. 5, pp. 834, 2021, DOI: 10.3390/pr9050834.
14. Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019, DOI: 10.1109/ACCESS.2019.2904620.
15. Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion Detection Using Convolutional Neural Networks for Representation Learning," in *Neural Information Processing. ICONIP 2017*, Lecture Notes in Computer Science, vol. 10638, D. Liu, S. Xie, Y. Li, D. Zhao, and E. S. El-Alfy, Eds., Springer, Cham, 2017, pp. 858–866, DOI: 10.1007/978-3-319-70139-4_87.
16. C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, DOI: 10.1109/ACCESS.2017.2762418.
17. M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1185–1190, 2012, DOI: 10.1007/s00521-010-0487-0.
18. I.S.Thaseen et al., "An integrated intrusion detection system using correlation-based attribute selection and artificial neural network," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, 2021, Art. no. e4014, DOI: 10.1002/ett.4014.

19. M.Ramaiah et al., “An intrusion detection system using optimized deep neural network architecture,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, 2021, Art. no. e4221, DOI: 10.1002/ett.4221.

20. C. Xu et al., “An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units,” *IEEE Access*, vol. 6, pp. 48697–48707, 2018, DOI: 10.1109/ACCESS.2018.2867564.

21. T.-T.-H. Le, Y. Kim, and H. Kim, “Network Intrusion Detection Based on Novel Feature Selection Model and Various Recurrent Neural Networks,” *Applied Sciences*, vol. 9, no. 7, pp. 1392, 2019, DOI: 10.3390/app9071392.

22. A.Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, “A survey of neural networks usage for intrusion detection systems,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 497–514, 2021, DOI: 10.1007/s12652-020-02014-x.

23. H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” *Applied Sciences*, vol. 9, no. 20, 2019, Art. no. 4396, DOI: 10.3390/app9204396.

24. A.I.Getman et al., “Deep Learning Applications for Intrusion Detection in Network Traffic,” *Proceedings of ISP RAS*, vol. 35, no. 4, pp. 65–92, 2023.

25. CICIDS2017 Dataset. [Online]. Available: https://www.unb.ca/cic/datasets/ids2017.html [Accessed: Aug. 16, 2020].

Information about authors:

Altynay Serdaly (corresponding author) – Master student of the Al-Farabi Kazakh National University, Faculty of Information Technology (Almaty, Kazakhstan, e-mail: altynayserdaly@gmail.com).

Dana Imanbekkyzy – Master student of the Al-Farabi Kazakh National University, Faculty of Information Technology (Almaty, Kazakhstan, e-mail: Danaimanbekkyzy01@gmail.com).

Adil Akhmetay – Master student of the Al-Farabi Kazakh National University, Faculty of Information Technology (Almaty, Kazakhstan, e-mail: akhmetay.adil@gmail.com)

Batyrkhan Omarov – PhD, International Information Technology University, Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: batyahan@gmail.com).