UDC 004.738.7
DOI: 10.15587/1729-4061.2025.326779

FORECASTING ANOMALIES IN NETWORK TRAFFIC

Inkar Zhumay
Doctoral Student*
Kymyssay Tumanbayeva
Candidate of Technical Sciences, Professor*
Katipa Chezhimbayeva
Corresponding author
Candidate of Technical Sciences, Professor*
E-mail: k.chezhimbayeva@aues.kz
Kuat Kalibek
Master of Science in Engineering*
*Department of Telecommunication Engineering
Non-Profit Joint Stock Company "Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev"
Baytursynuli str., 126/1, Almaty, Republic of Kazakhstan, 050013

The increasing volume of traffic, growing number of connections in telecommunication networks, and rising number of mobile devices place significant demands on network providers. These challenges can lead to congestion, latency issues, and security vulnerabilities. However, they can be mitigated or even prevented by identifying network failures in advance. Anomaly detection plays a crucial role in proactively addressing these issues, enabling network operators to optimize network performance, enhance security, and improve the overall user experience.

In this study, a method for predicting anomalies based on machine learning has been implemented. The LSTM-SMOTE model, which was trained and tested on the KDD-NLS dataset, was considered and the results of the forecasting model were analyzed. Developing the multi-classification model proved to be a challenging task, primarily due to the limited number of attack types. SMOTE is designed to address such difficulties. Imbalanced datasets present a major challenge in predictive modeling, especially when solving classification problems. The four main types of attacks include Denial of service (DoS) attacks, Probe attacks, Privilege attacks, and Access attacks. In this work, three neural network models were developed, including: binary classification, four-class classification, multi-class classification. It is observed that the prediction model retrained again showed the best results, then the model trained with new anomalous data. The LSTM-SMOTE multiclass model achieved the highest performance, with its predictive accuracy rising from 75 % to 99 % across iterations, underscoring its strong dependence on the quality and quantity of data. Practical application of the results obtained can be applied for optimizing network performance

Keywords: Network anomaly, telecommunication traffic, network traffic prediction, long short-term memory (LSTM), semi-supervised learning

Received 30.01.2025
Received in revised form 11.03.2025
Accepted date 08.04.2025
Published date 22.04.2025

How to Cite: *Zhumay, I., Tumanbayeva, K., Chezimbayeva, K., Kalibek, K. (2025). Forecasting anomalies in network traffic. Eastern-European Journal of Enterprise Technologies, 2 (2 (134)), 96–111. https://doi.org/10.15587/1729-4061.2025.326779*

1. Introduction

Contemporary networks, such as 5G, IoT, cloud, and hybrid infrastructures, produce vast amounts of heterogeneous traffic. Conventional monitoring techniques are increasingly inadequate for real-time analysis and detecting complex threats.

In the face of rapidly increasing network traffic, expanding digital infrastructure, and the widespread integration of IoT, cloud, and mobile technologies, ensuring the resilience and security of telecommunications networks has become more critical than ever. A central focus in this domain is the prediction of anomalies in network traffic, which acts as an essential mechanism for the early detection of potential threats and deviations in data transmission systems.

Traditional monitoring and protection methods based on signatures or static rules are proving to be insufficiently effective in the face of increasingly sophisticated and stealthy attacks, often disguised as legitimate activity. Moreover, as network architectures grow in scale and complexity, the risk of unforeseen failures, overloads, and other deviations not directly linked to malicious activity but significantly impacting the quality and security of provided services also increases.

Scientific research in the field of anomaly prediction enables the development of adaptive models capable of analyzing traffic behavior in real time, detecting deviations from the norm, and forecasting the development of potentially dangerous situations before they actually occur. The use of machine learning and data mining methods is particularly important, as they can automatically adapt to changing conditions and types of threats.

The results of such research have wide practical applications: from enhancing the efficiency of information security systems (SOC, SIEM) to optimizing network resources, minimizing downtime, and reducing service unavailability. Additionally, they help to develop domestic technological solutions in the field of telecommunications cybersecurity.

Therefore, the research on anomaly prediction methods in network traffic holds not only scientific and technical value but also considerable practical importance for the advancement of a stable and secure digital infrastructure.

2. Literature review and problem statement

The research in article [1] on Support Vector Regression (SVR) and Neighbor Coordination has shown promising outcomes; however, some challenges remain unaddressed. Notably, the algorithm is specifically designed for meteorological elements, restricting its broader applicability, and it does not include comparisons with other classifier types. These limitations may arise due to the distinctive nature of meteorological data, the complexity of incorporating multiple classifiers, or the high computational cost of comprehensive comparisons. A potential way to address these challenges is by conducting evaluations against multiple classifiers to ensure broader validation.

The study on LSVM in work [2] has demonstrated considerable potential, yet some limitations persist. In particular, the method struggles with handling unsupervised data, limiting its usability in scenarios without labeled datasets. This limitation may be due to the inherent nature of LSVM, which relies on supervised learning, or the challenges associated with adapting it for unsupervised tasks. One possible approach to addressing these challenges is refining the technique for anomaly detection, especially for detecting link failures.

The study in paper [3] presented a machine learning approach for automatic anomaly prediction in SDN, utilizing traditional machine learning techniques. This method classifies network traffic as either normal or anomalous before applying anomaly prevention strategies. However, their work did not specifically address the challenge of detecting minor false positives and false negatives, improving the dataset and classification methodology, or advancing anomaly classification with greater granularity for more effective control policies.

In work [4], researchers developed a DPI system incorporating traffic analysis algorithms, Hurst parameters, and the three-sigma method for anomaly detection. Their approach was compared with the existing DPS SolarWinds system. However, the study did not specifically address the challenge of extending the proposed DPI system to detect other types of attacks.

In [5], different approaches to traffic forecasting with artificial neural networks (ANNs) are examined. The study found that, in some cases, simpler models like RNB and MLP can outperform more complex ones such as SAE. The role of deep neural networks is becoming increasingly significant, achieving remarkable success in pattern recognition tasks involving images, audio, and video. However, most learning algorithms for these networks rely on unsupervised training, utilizing unlabeled data. In contrast, network traffic and time series data are generally labeled, requiring an initial unsupervised pre-training phase before applying supervised tuning.

In [6], a hybrid method was proposed for the early detection of abnormal traffic, aiming to assist network administrators in taking quick preventive actions to ensure a more stable and accessible network. However, the study did not specifically address the challenge of reducing the false negative rate by enhancing and expanding MLP training on normal traffic. Moreover, the impact of employing a dynamic window size for the control chart has yet to be investigated.

In [7], proposed approach also considers that classifier accuracy can be optimized by applying dataset balancing or re-sampling methods. Further research needs to be conducted using deep learning methods.

In [8], an architecture for a flexible anomaly detection system is proposed, using a modular approach and the principle of scalability to enhance efficiency in networks. However, the study did not specifically tackle the challenge of analyzing protocols beyond TCP or detecting a broader range of attacks. Additionally, while re-training methods and source detection were discussed, they require further exploration, along with comparative analyses involving unsupervised techniques.

In [9], a study was conducted that the increasing traffic volume has driven an increased demand for reliable mobile communication systems, highlighting the key role of wearable applications in various sectors including biomedical, military and rescue services, with a special focus in this study on the design and development of wearable antennas for Internet of Things (IoT) applications for efficient traffic management.

The study [10] focuses on reducing network congestion in gateways by applying teletraffic theory to optimize critical performance metrics, including latency, throughput, and scalability. However, it highlights the necessity of real-world experiments to address challenges related to device heterogeneity and compatibility.

The paper [11] introduces a traffic simulation model for short-distance transmission networks, illustrating the connection between network server buffer memory and packet loss probability under the influence of incoming self-similar traffic characteristics. This highlights the crucial role of high-quality programs as a foundation for system reliability.

Ensuring high reliability of telecommunications systems is a key factor in maintaining the stable and secure operation of network infrastructure. Improving the reliability of telecommunication infrastructure is impossible without a thorough analysis of network traffic, which helps to uncover hidden anomalies and vulnerabilities. Moreover, most of the published results regarding reliability growth primarily focus on software reliability. However, the growth of hardware reliability resulting from the elimination of design errors has been rarely studied.

As a result of the analysis of scientific publications, it has been established that despite the active development of anomaly detection methods in network traffic, most works focus on retrospective analysis of already recorded incidents. A significant portion of existing solutions requires considerable computational resources, high-quality data labeling, and demonstrates insufficient resilience when network conditions change. Furthermore, existing studies show a lack of approaches that account for the temporal structure of traffic and are aimed at proactive anomaly detection, that is, predicting potentially dangerous states before they occur.

Thus, a common unresolved issue can be identified: the lack of effective and adaptive methods for anomaly prediction in network traffic, ensuring high accuracy under changing conditions and with limited prior information about potential threats.

All of this suggests that it is reasonable to conduct a study on the analysis and optimization of the predictive method for detecting network traffic anomalies.

3. The aim and objectives of the study

The aim of this study is to analyze and optimize a predictive method for detecting network traffic anomalies. The practical component highlights the anticipated benefits of applying the results in real-world scenarios. This approach will enable more accurate and timely identification of traffic disruptions, facilitating proactive traffic management and optimization. The practical application of the obtained results can contribute to optimizing network performance, ensuring more reliable and efficient traffic flow management.

To achieve this aim, the following objectives are accomplished:

– to implement and compare different neural network models binary, 4-class, and multiclass classification the study aims to enhance the accuracy of anomaly detection in network traffic;

– to enhance the SMOTE-LSTM model for the analysis of telecommunication networks to strengthen its capability in addressing class imbalance and increasing predictive accuracy;

– to enhance the accuracy of the SMOTE-LSTM method by retraining multi-classification models.

4. Materials and methods

The object of the study is the time series of network traffic parameters, which reflect changes in network characteristics over time, including indicators of load imbalance, the number of connections, data transmission volumes, and other metrics that potentially indicate abnormal behavior. The subject of the study is the methods for analyzing and forecasting anomalies based on such time series, aimed at proactive detection of potential failures and threats, including cyberattacks and network overloads.

The aim of this study is to analyze and optimize the predictive method for detecting network traffic anomalies, based on considering the temporal structure and dynamics of network parameter changes. The main hypothesis is that temporal dependencies in network traffic behavior allow for effective prediction of abnormal states, and the use of adaptive models that account for traffic dynamics contributes to increased accuracy and timeliness in detecting potential failures and attacks.

To achieve the set aim, the study implements a set of objectives aimed at improving the accuracy of prediction and the robustness of models to changing conditions. Specifically, the study involves the implementation and comparative analysis of various neural network models, including binary, four-class, and multi-class classification, with the aim of enhancing the effectiveness of anomaly detection in network traffic. Special attention is given to the improvement of the SMOTE-LSTM model, designed to work with imbalanced data typical of real telecommunications systems.

Approaches to retraining multi-class models are proposed, aimed at improving the accuracy and robustness of predictions in conditions of incomplete prior information and high traffic dynamics.

The research assumes that there are statistically significant features distinguishing normal from anomalous traffic behavior, that models can be trained without precise manual data labeling, and that aggregated time series data is sufficient for identifying patterns. For simplification, the analysis is confined to network metadata, excluding packet content inspection and disregarding external influences such as configuration changes or hardware malfunctions.

Network traffic data is taken from the database used for the third international competition held by KDD-99 in collaboration with the fifth International Conference on Data Mining. This database contains standard validation datasets that cover a wide range of accidents modeled in a network environment.

The first major disadvantage of the KDD dataset is the large number of duplicate records. By analyzing the KDD training and testing datasets, it is possible to found that approximately 78 % and 75 % of the records are repeated in the training and testing sets, respectively. This large number of duplicate records in the training set causes learning algorithms to be more biased toward the more frequent records and, as a result, they are unable to explore rare records, which are usually dangerous for networks.

To resolve the issues, all duplicate entries were removed from the training and test sets. Records from value groups were selected randomly to create a more complex subset of the KDD datasets, so that the number of records selected from each group is inversely proportional to the percentage of records in the original value groups. For example, the number of records in the 0–5 value group of the KDD training set constitutes 0.04 % of the original records, so 99.96 % of the records in this group are assigned to the established model. The created KDDTrain+ and KDDtest+ datasets included 125,973 and 22,544 records, respectively. Additionally, another test set was created, which included 11,850 records.

This part contains about 126,000 data records. Each set has 42 attributes, with only 3 discrete and 1 conditional being necessary.

The program is divided into three files:

a) main – the main file, where the core operations take place;

b) funcDef – a set of functions that contain the algorithms performing statistical analysis;

c) KDDPreprocessing – this is the file that processes the dataset into a state that can be read by our program.

In the field of network telecommunications, there are several types of emergencies that threaten network security. The four main types of attacks include Denial of service (DoS) attacks, Probe attacks, Privilege attacks, and Access attacks.

5. Research results based on the analysis of anomaly prediction methods in time series for forecasting anomalies in network traffic

5. 1. Implementation of network traffic classification models

With the growth of network traffic and the increasing number of cyber threats, the task of timely anomaly detection and prediction becomes critically important. Anomalies may indicate DDoS attacks, intrusions, or configuration errors. This paper examines existing approaches and presents a machine learning-based forecasting method.

In future research, three neural network models binary classification, four-class classification, and multi-class classification will be used for anomaly detection in network traffic.

Within the four-class classification approach, the model demonstrates a strong ability to distinguish between DoS, Probe, Privilege, and Access attack types. Classification accuracy and reliability metrics have confirmed the model's high effectiveness in identifying these attack types, contributing to improved network security monitoring.

Applying the multi-class classification model has improved the accuracy and detail of network attack detection, providing a deeper understanding of threat types and enabling proactive response to them.

The SMOTE-LSTM model (Synthetic Minority Over-sampling Technique+Long Short-Term Memory) is a combination of two powerful methods: SMOTE for handling imbalanced data and LSTM for time series analysis, making it effective for telecommunication network analysis, particularly for anomaly detection and attack classification.

The SMOTE-LSTM model outperforms traditional 4-class classification in handling imbalanced data, accounting for temporal dependencies, and detecting anomalies, offering higher classification accuracy and improved recognition of rare attacks. Therefore, in the future, let's apply the SMOTE-LSTM model for anomaly detection.

The multi-class classification model can significantly enhance anomaly detection by distinguishing a broader range of anomalous traffic types. Despite the increased classification complexity, which may reduce accuracy in certain attack categories, the proposed model will provide valuable insights for proactive threat mitigation (Fig. 1).

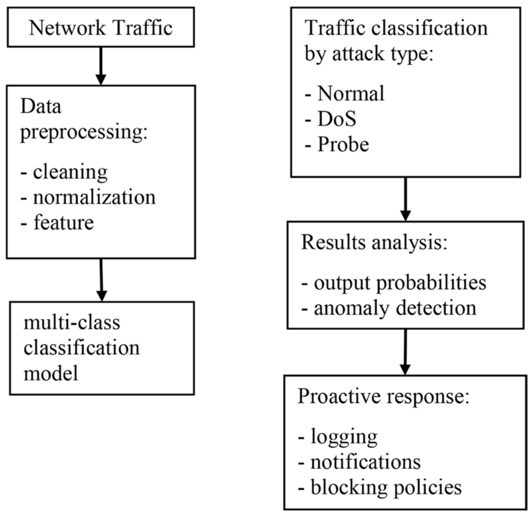


Fig. 1. Structural block diagram of the multi-class network traffic classification model

Applying the multi-class classification model has improved the accuracy and detail of network attack detection, providing a deeper understanding of threat types and enabling proactive response to them.

5. 2. SMOTE-LSTM model for telecommunication network analysis

The SMOTE-LSTM model follows a two-step approach. First, SMOTE is applied to the dataset to address class imbalance by synthetically generating new instances of minority classes. This ensures that the model does not favor the majority class, leading to a more balanced classification result. Then, the oversampled data is fed into the LSTM network, which is well-suited for learning from sequential and time-dependent data. The LSTM layers capture long-term dependencies and extract meaningful patterns from the input data, improving classification accuracy.

The implementation includes the following steps:

– data preprocessing: the dataset is cleaned, normalized, and split into training and test sets;

– oversampling with SMOTE: SMOTE is applied to the training set, creating synthetic examples for underrepresented classes;

– building the LSTM model: a deep LSTM network is designed with multiple layers, including an embedding layer, LSTM layers, dropout layers for regularization, and a dense output layer with softmax activation for multi-class classification;

– training and optimization: the model is trained using backpropagation through time (BPTT) and optimized with Adam or RMSprop optimizers. Performance is evaluated using accuracy, precision, recall, and F1-score metrics.

The method consists of several steps, as shown in Fig 2.

Example of a model: LSTM (Long Short-Term Memory) for network traffic time series forecasting.

LSTM model architecture:

Input series: X = [t1, t2, ..., tn] — traffic measurements (e.g., bytes per second).

Input Layer

LSTM layer (or multiple)

Dropout (for regularization)

Dense layer

Output: prediction of the value for the next time point (t+1)

→ Error = |prediction - actual value|

→ If error > threshold → anomaly is recorded.

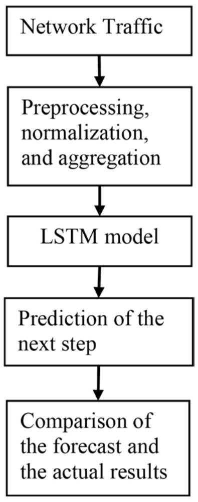

Network Traffic
Preprocessing, normalization, and aggregation
LSTM model
Prediction of the next step
Comparison of the forecast and the actual results

Fig. 2. General block diagram of the forecasting+detection model

For example, using a simple model in Keras for network traffic forecasting with LSTM, which can be used for anomaly detection. This study includes error handling and anomaly detection.

As a result of the analysis, it is possible to determine whether the traffic is normal or anomalous, and if anomalous, the type of failure. The main types of DDoS attacks are Probe, Access, and Privilege. The architecture is as follows: there is a folder containing the analysis code, neural network, and result images. The architecture is shown in Fig. 2.

5. 3. The implementation of the SMOTE-LSTM model by retraining multi-classification models

In this paper, 6 scenarios are considered:

a) binary classification, training and testing from a single dataset;

b) binary classification, testing and training from two different datasets;

c) 4-dimensional target classification, testing and training from a single dataset;

d) 4-dimensional target classification, testing and training from two different datasets;

e) multiplicative classification, testing and training from a single dataset;

f) a scenario where the test dataset is used for training and the training dataset is used for testing.

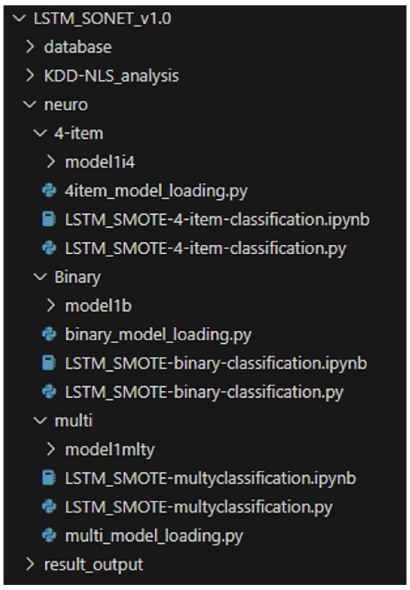
Well-prepared data is needed for effective training of neural networks. This involves several key steps: cleaning and preprocessing the data, splitting it into training and testing sets, and organizing them into features (*X*) and labels (*Y*). These steps ensure that the model can explore patterns in the training set and generalize them well to unseen data in the test set.

Features (*X*): input data that the model uses to learn patterns.

Labels (*Y*): the target variable that the model tries to predict.

The KDD-NLS dataset and the Python programming language were used for the analysis. The main Python libraries used in the analysis are Pandas, TensorFlow, Keras, Sklearn, and Matplotlib.

As a result of the analysis, it is possible to determine whether the traffic is normal or anomalous, and if it is anomalous, the type of failure. The main types of DDoS attacks are Probe, Access, Privilege. The architecture is as follows: there is a folder where the analysis code, neural network, and result images are stored. The architecture is shown in Fig. 3.



```
∨ LSTM_SONET_v1.0
  > database
  > KDD-NLS_analysis
  ∨ neuro
    ∨ 4-item
      > model1i4
        4item_model_loading.py
        LSTM_SMOTE-4-item-classification.ipynb
        LSTM_SMOTE-4-item-classification.py
    ∨ Binary
      > model1b
        binary_model_loading.py
        LSTM_SMOTE-binary-classification.ipynb
        LSTM_SMOTE-binary-classification.py
    ∨ multi
      > model1mlty
        LSTM_SMOTE-multyclassification.ipynb
        LSTM_SMOTE-multyclassification.py
        multi_model_loading.py
  > result_output
```

Fig. 3. Architecture

At the beginning, let's import the software libraries that it is possible to use in the work. Let's load the required datasets into the program and transform them into a format suitable for processing. These include:

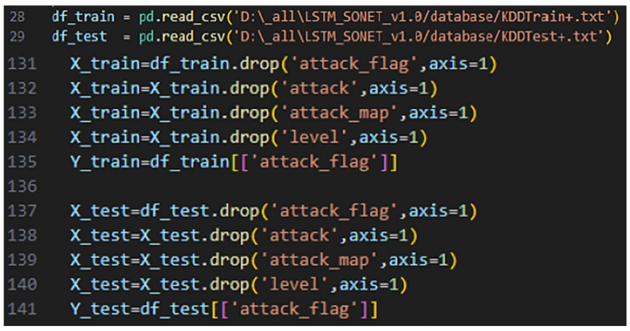
1. Converting the data types of protocol_type, service, and flag into discrete form.

2. Converting the column with the actual names of failures into a discrete type and divide it into normal/anomalous categories.

Let's prepare the test and training data and convert the data into discrete form and split it into columns *X* and *Y* in Fig. 4. Part of the performed data recovery and auxiliary processes demonstrated in a Fig. 5.

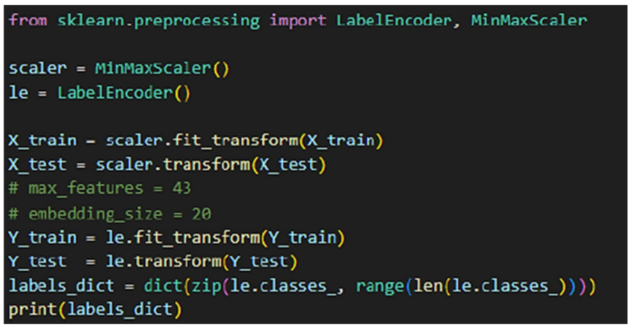
The resulting dataset includes standard ML processes such as normalization and recovery of missing values. Next, let's adjust the form of data storage so that our neural network can read the input data demonstrated in a Fig. 6.

In data science and machine learning, methods such as Fit(), transform(), and fit_transform(), offered by the scikit-learn package, are among the most important tools widely used in data preprocessing.



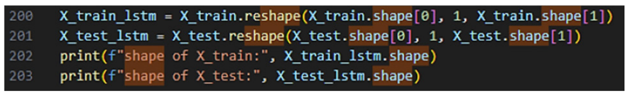
```python
df_train = pd.read_csv('D:\_all\LSTM_SONET_v1.0/database/KDDTrain+.txt')
df_test  = pd.read_csv('D:\_all\LSTM_SONET_v1.0/database/KDDTest+.txt')
  X_train=df_train.drop('attack_flag',axis=1)
  X_train=X_train.drop('attack',axis=1)
  X_train=X_train.drop('attack_map',axis=1)
  X_train=X_train.drop('level',axis=1)
  Y_train=df_train[['attack_flag']]

  X_test=df_test.drop('attack_flag',axis=1)
  X_test=X_test.drop('attack',axis=1)
  X_test=X_test.drop('attack_map',axis=1)
  X_test=X_test.drop('level',axis=1)
  Y_test=df_test[['attack_flag']]
```

Fig. 4. X and Y, distribution and preparation of test and training data



```python
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

scaler = MinMaxScaler()
le = LabelEncoder()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# max_features = 43
# embedding_size = 20
Y_train = le.fit_transform(Y_train)
Y_test  = le.transform(Y_test)
labels_dict = dict(zip(le.classes_, range(len(le.classes_))))
print(labels_dict)
```

Fig. 5. Part of the performed data recovery and auxiliary processes



```python
X_train_lstm = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test_lstm = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
print(f"shape of X_train:", X_train_lstm.shape)
print(f"shape of X_test:", X_test_lstm.shape)
```

Fig. 6. Changing the form of data storage

There are the following steps remaining:

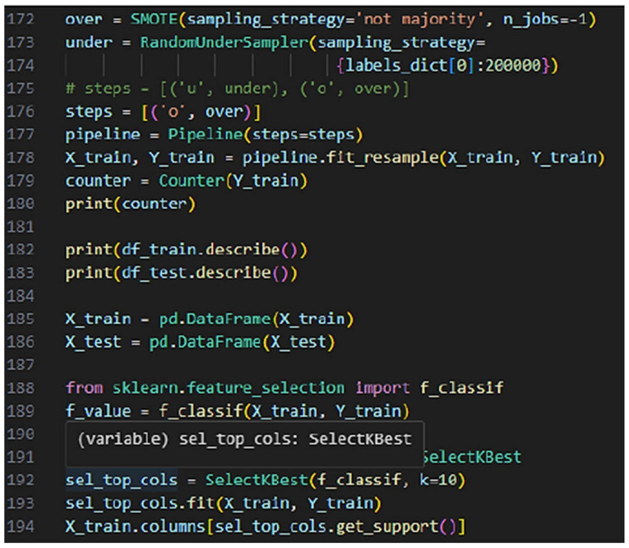
a) using SMOTE;

b) defining the LSTM model;

c) compiling and training the model: using binary cross-entropy loss and the Adam optimizer to train the model.

Smote Implementation demonstrated in a Fig. 7.

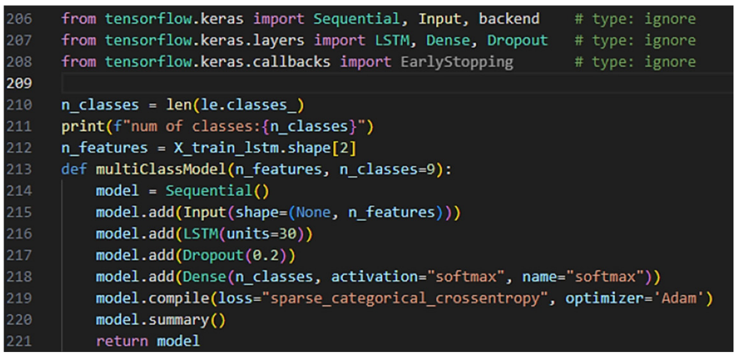
Next, let's save the standard TensorFlow function as cp1 (from the word checkpoint).



```python
over = SMOTE(sampling_strategy='not majority', n_jobs=-1)
under = RandomUnderSampler(sampling_strategy=
                           {labels_dict[0]:200000})
# steps - [('u', under), ('o', over)]
steps = [('o', over)]
pipeline = Pipeline(steps=steps)
X_train, Y_train = pipeline.fit_resample(X_train, Y_train)
counter = Counter(Y_train)
print(counter)

print(df_train.describe())
print(df_test.describe())

X_train - pd.DataFrame(X_train)
X_test = pd.DataFrame(X_test)

from sklearn.feature_selection import f_classif
f_value = f_classif(X_train, Y_train)
 (variable) sel_top_cols: SelectKBest
                                        SelectKBest
sel_top_cols = SelectKBest(f_classif, k=10)
sel_top_cols.fit(X_train, Y_train)
X_train.columns[sel_top_cols.get_support()]
```

Fig. 7. Smote implementation

Let’s build the model using a function that creates the LSTM model. It inputs the shape of our input data into the model and starts the model training process. Within the standard parameters that are added when creating the model (apart from the input shape), the model type (in our case, Sequential model), the LSTM model (which takes the number of neural network layers as a parameter), the Adam optimizer, and the Dropout and Dense layers are included. During the model training process, let’s provide the *X* and *Y* sets of our training dataset. Additionally, let’s specify the number of iterations for the model, and input parameters such as validation, optimizer, and the memory device. Automation function for creating an LSTM model presented in a Fig. 8.



```
from tensorflow.keras import Sequential, Input, backend    # type: ignore
from tensorflow.keras.layers import LSTM, Dense, Dropout   # type: ignore
from tensorflow.keras.callbacks import EarlyStopping       # type: ignore

n_classes = len(le.classes_)
print(f"num of classes:{n_classes}")
n_features = X_train_lstm.shape[2]
def multiClassModel(n_features, n_classes=9):
    model = Sequential()
    model.add(Input(shape=(None, n_features)))
    model.add(LSTM(units=30))
    model.add(Dropout(0.2))
    model.add(Dense(n_classes, activation="softmax", name="softmax"))
    model.compile(loss="sparse_categorical_crossentropy", optimizer='Adam')
    model.summary()
    return model
```

Fig. 8. Automation function for creating an LSTM model

Building and training the model.

In this work, since the device has limited power, the number of training iterations is set to 500, and the number of neural layers in the LSTM model is 50.

The data is divided into three types:

1. nr-met\at-Met # NR – normal;
2. nr-hit\at-type # at-attack;
3. nr-traffic\attack # traf-traffic.

The forecast results are mainly displayed using the TPFP graph and the ROC Curve, and the values underlying the ROC Curve are shown. Key graphs, such as the true positive (TP), false positive (FP), true negative (TN), false negative (FN) curves, and the receiver operating characteristic (ROC) curve, are crucial for evaluating the performance of classification models. These metrics and visualizations help assess how well the model distinguishes between different classes, especially in binary classification tasks.

True positive occurs when the model correctly predicts the positive class. For example, the incoming traffic was normal, and our model correctly identified it as real. False positive occurs when the model incorrectly predicts the positive class. If the program considers normal traffic as an attack, it is classified as a false positive (FP). Additionally, in IDS systems, this is referred to as a false alarm. Considering this data, along with TN and FN, and converting it into a graphical format, let’s get the Confusion Matrix.

The ROC curve is a graphical representation that shows the diagnostic ability of a binary classifier as it varies with different discrimination thresholds.

Understanding true positives, false positives, and ROC curves is fundamental for evaluating the performance of classification models.

Fig. 9 shows the dependence of data loss values during model training on the number of iterations. This result is considered an indicator only for the studied values. The model’s response to new types of accidents is usually different.

After the prediction, PNFN calculates the Confusion Matrix and prepares functions that generate the ROC curve. With their help, the following graphs are constructed, Binary TPFP Train & Test Dataset are shown in Fig. 10. LSTM-SMOTE Binary Classification ML analysis is presented in Fig. 11.

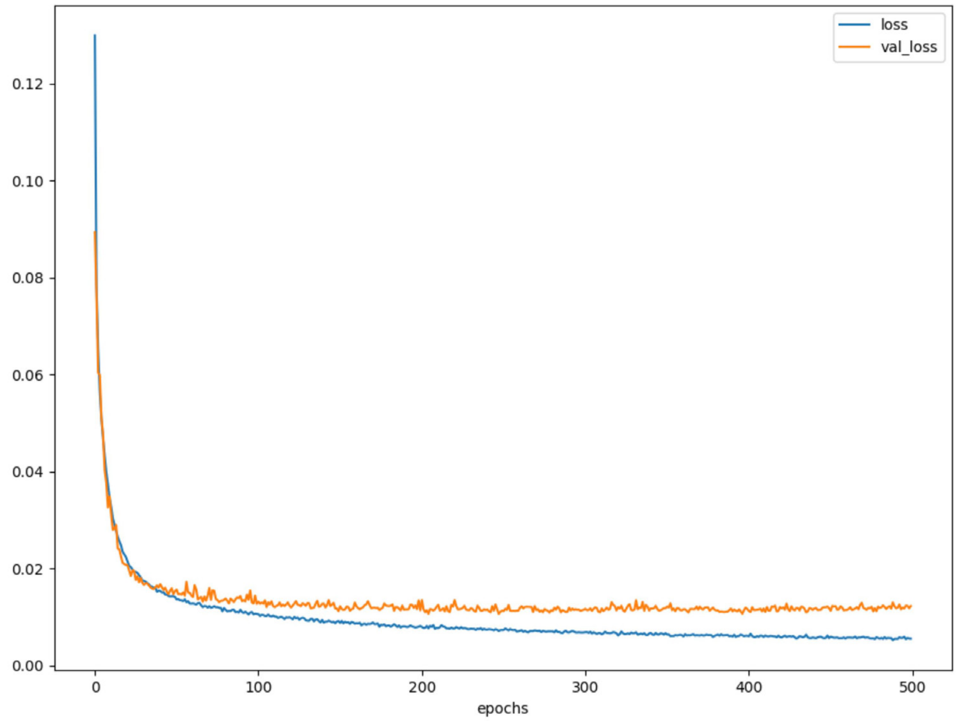
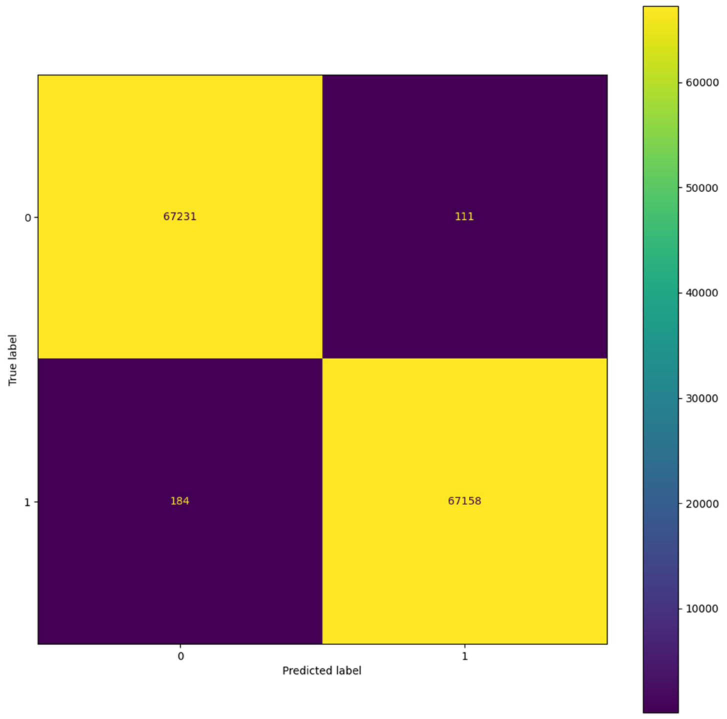
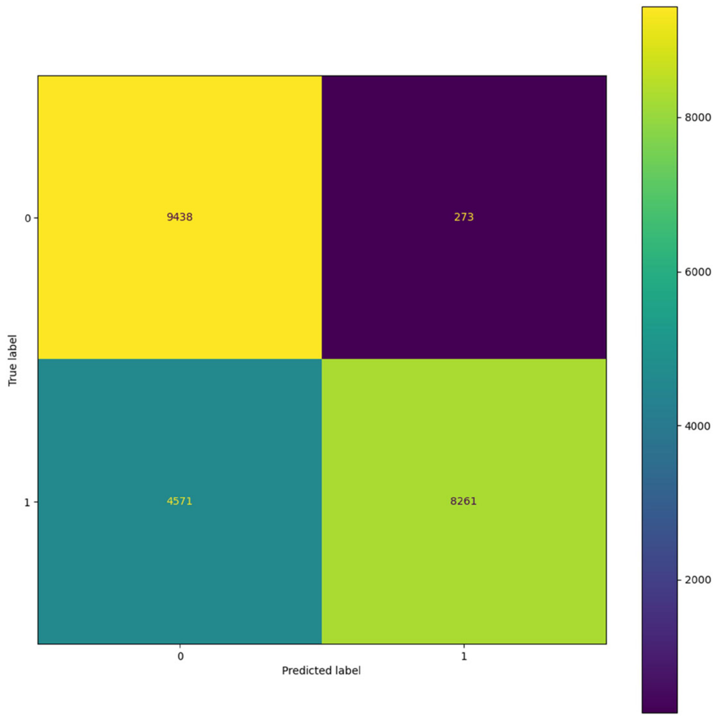


Fig. 9. Binary model. Model training results



a

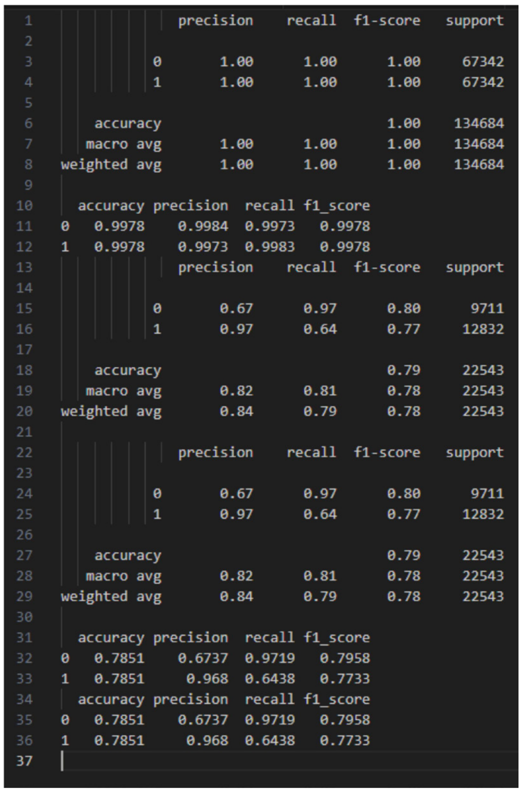


b

Fig. 10. Binary value: *a* — train dataset; *b* — test dataset

This window shows the result of analyzing the forecast results of training and testing with ML parameters. In addition to the standard accuracy, precision, recall, there is also an f score.

In statistical analysis of binary classification and information retrieval systems, the F-score or F-measure is a performance metric for prediction. It is calculated based on precision and recall.



```
                precision    recall  f1-score   support

           0       1.00      1.00      1.00     67342
           1       1.00      1.00      1.00     67342

    accuracy                           1.00    134684
   macro avg       1.00      1.00      1.00    134684
weighted avg       1.00      1.00      1.00    134684

   accuracy precision   recall f1_score
0    0.9978    0.9984  0.9973   0.9978
1    0.9978    0.9973  0.9983   0.9978
                precision    recall  f1-score   support

           0       0.67      0.97      0.80      9711
           1       0.97      0.64      0.77     12832

    accuracy                           0.79     22543
   macro avg       0.82      0.81      0.78     22543
weighted avg       0.84      0.79      0.78     22543

                precision    recall  f1-score   support

           0       0.67      0.97      0.80      9711
           1       0.97      0.64      0.77     12832

    accuracy                           0.79     22543
   macro avg       0.82      0.81      0.78     22543
weighted avg       0.84      0.79      0.78     22543

   accuracy precision   recall f1_score
0    0.7851    0.6737  0.9719   0.7958
1    0.7851     0.968  0.6438   0.7733
   accuracy precision   recall f1_score
0    0.7851    0.6737  0.9719   0.7958
1    0.7851     0.968  0.6438   0.7733
```

Fig. 11. LSTM-SMOTE binary classification ML analysis

As it is possible to see the testing results in the prediction of training data are very good. When it comes to the test data, it is possible to see that there is a noticeable difference between the values. ROC Curve: for Binary Train and Test are demonstrated in Fig. 12.

As the ROC analysis shows, our forecasting method has good results.

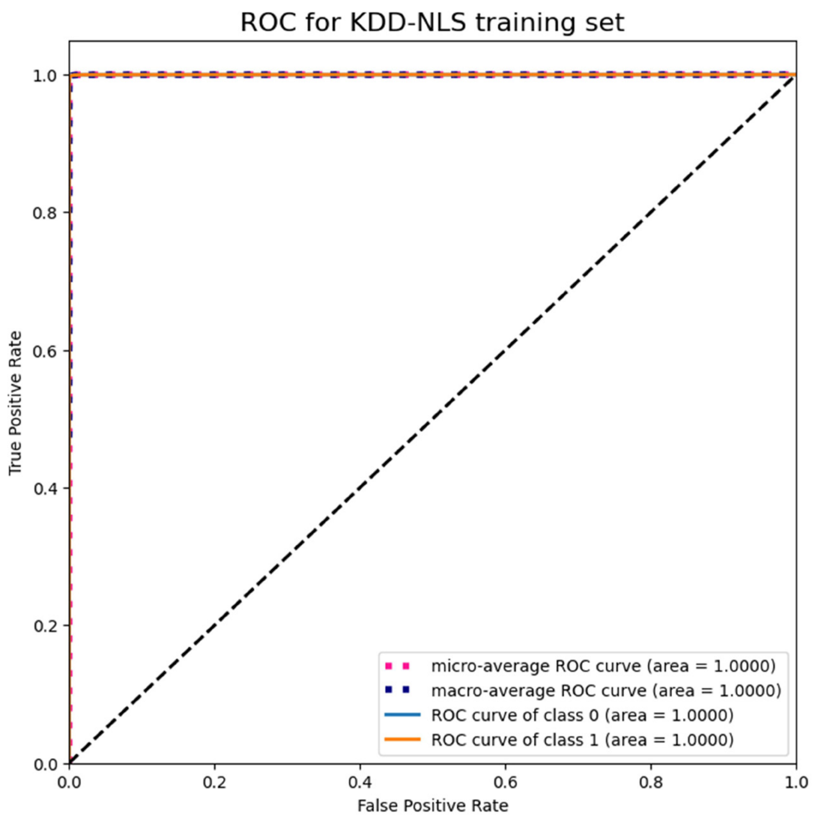
The graphs for four-element classification are displayed in Fig. 13, illustrating the forecast before training and the results after training.

This graph illustrates the TPFP results for the traffic type prediction using the LSTM-SMOTE model. The LSTM-SMOTE 4-item classification machine learning analysis is displayed in Fig. 14. The ROC curve for Binary Train and Binary Test is presented in Fig. 15.

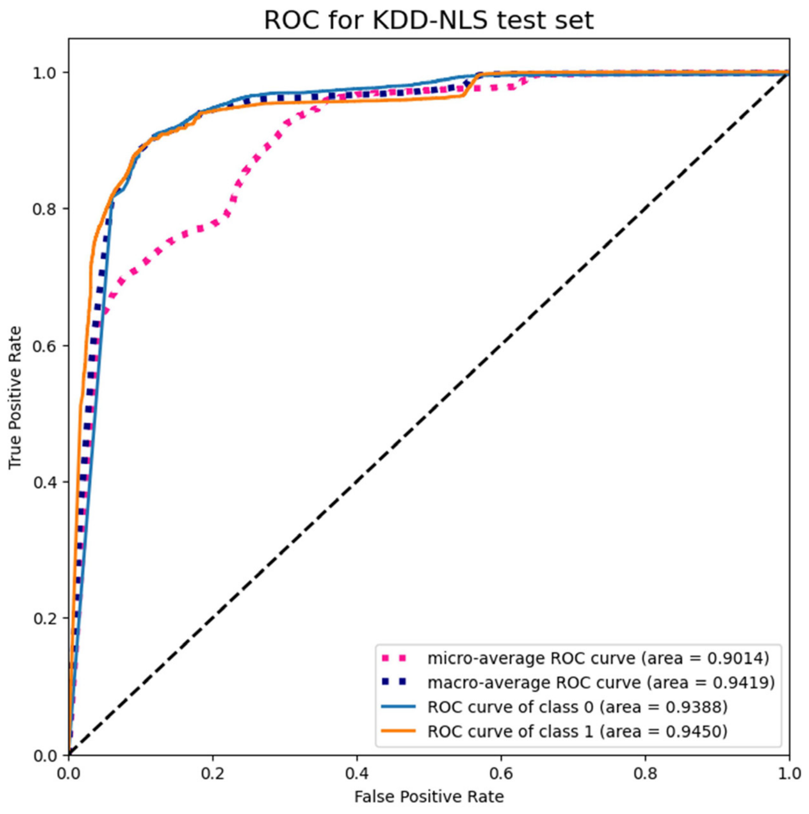
The multiclass classification model is shown in Fig. 16. The following graph was generated as a result of the training process:

Model training results. The figure shows the relationship between the loss values of the data during model training and the number of iterations. This result is considered an indicator only for the studied values. The model's response to new types of attacks is usually different.

After the prediction, PNFN calculates the Confusion Matrix and prepares functions that generate the ROC curve. Using these methods, the following graphs for Multi-classification TPFP are generated: Train Dataset and Test Dataset, Fig. 17.

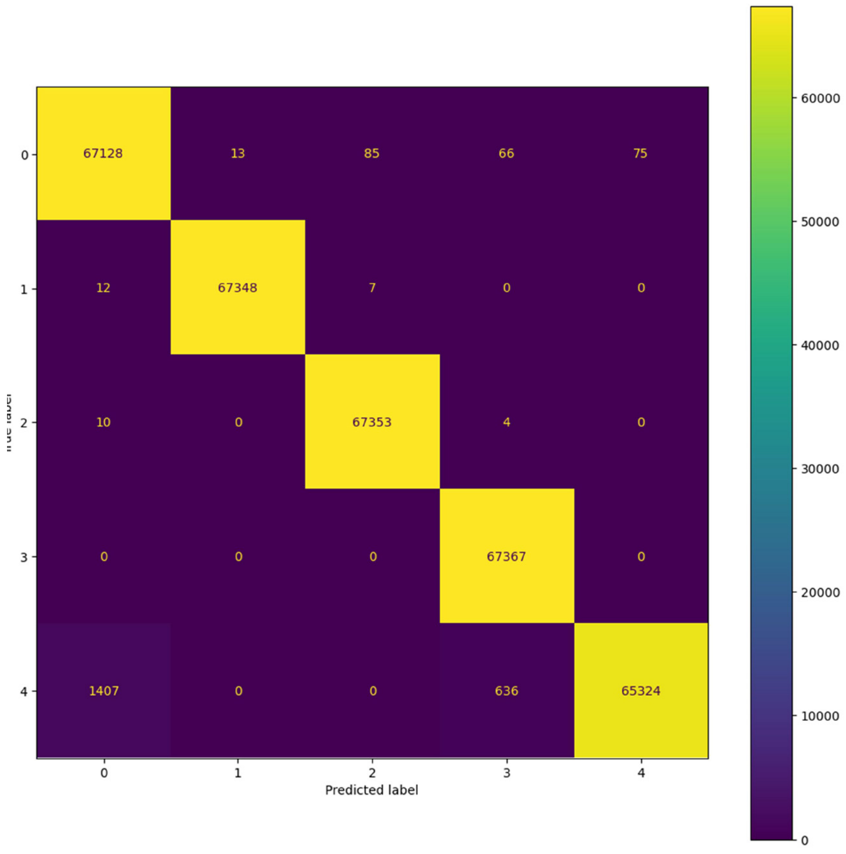


a

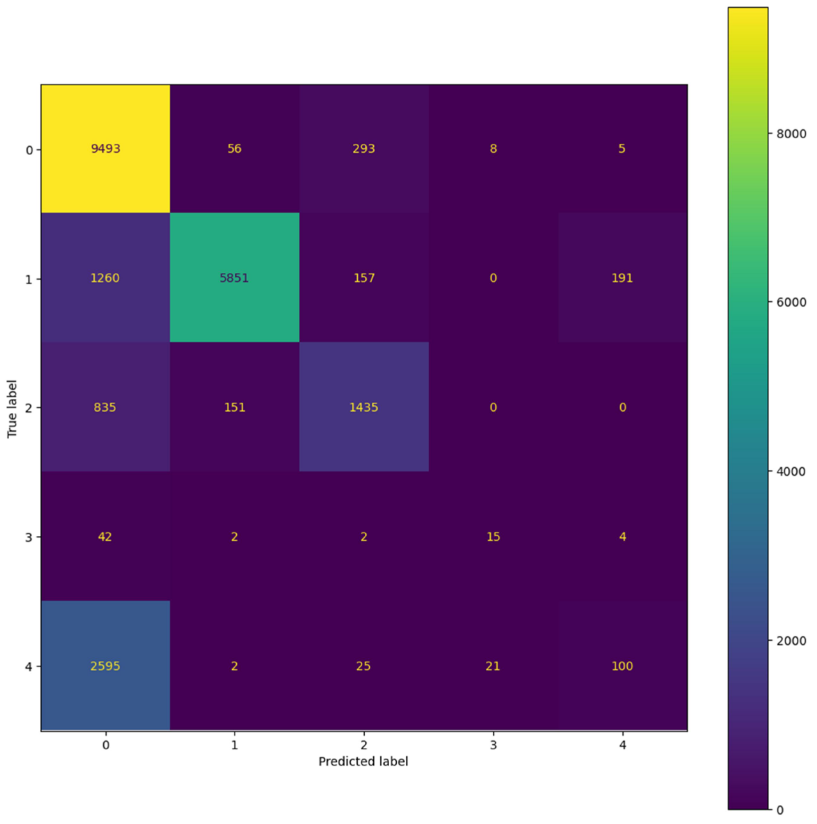


b

Fig. 12. Receiver operating characteristic (ROC) curve: *a* – binary train; *b* – binary test

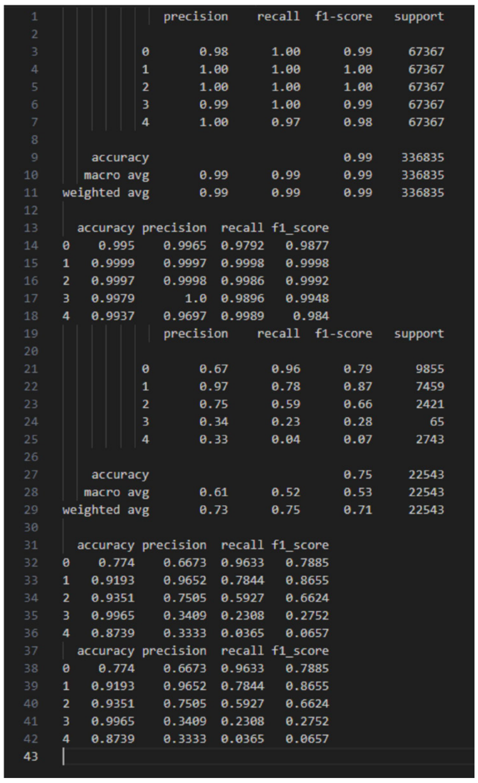


a



b

Fig. 13. 4-item True Positive, False Positive (TPFR): *a* – train dataset; *b* – test dataset



```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99     67367
           1       1.00      1.00      1.00     67367
           2       1.00      1.00      1.00     67367
           3       0.99      1.00      0.99     67367
           4       1.00      0.97      0.98     67367

    accuracy                           0.99    336835
   macro avg       0.99      0.99      0.99    336835
weighted avg       0.99      0.99      0.99    336835

  accuracy precision  recall f1_score
0    0.995    0.9965  0.9792   0.9877
1   0.9999    0.9997  0.9998   0.9998
2   0.9997    0.9998  0.9986   0.9992
3   0.9979       1.0  0.9896   0.9948
4   0.9937    0.9697  0.9989    0.984
              precision    recall  f1-score   support

           0       0.67      0.96      0.79      9855
           1       0.97      0.78      0.87      7459
           2       0.75      0.59      0.66      2421
           3       0.34      0.23      0.28        65
           4       0.33      0.04      0.07      2743

    accuracy                           0.75     22543
   macro avg       0.61      0.52      0.53     22543
weighted avg       0.73      0.75      0.71     22543

  accuracy precision  recall f1_score
0    0.774    0.6673  0.9633   0.7885
1   0.9193    0.9652  0.7844   0.8655
2   0.9351    0.7505  0.5927   0.6624
3   0.9965    0.3409  0.2308   0.2752
4   0.8739    0.3333  0.0365   0.0657
  accuracy precision  recall f1_score
0    0.774    0.6673  0.9633   0.7885
1   0.9193    0.9652  0.7844   0.8655
2   0.9351    0.7505  0.5927   0.6624
3   0.9965    0.3409  0.2308   0.2752
4   0.8739    0.3333  0.0365   0.0657
```

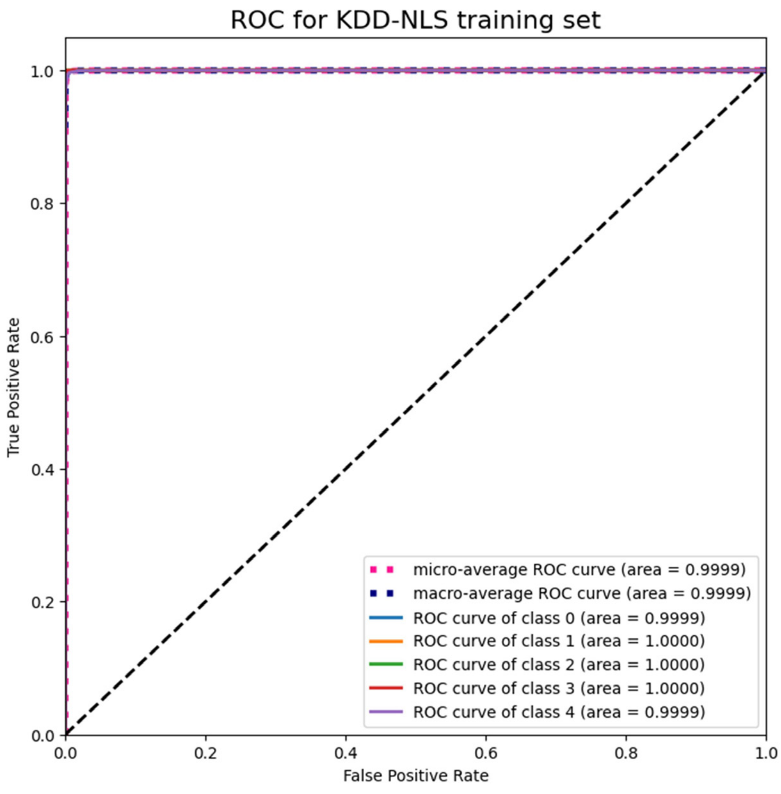
Fig. 14. LSTM-SMOTE 4-item classification ML analysis

This graph displays the TPFP results for traffic type prediction using the LSTM-SMOTE model, based on the test dataset. The exact numerical values are provided below. The ROC curve for multi-classification: Train and Test is shown in Fig. 18.

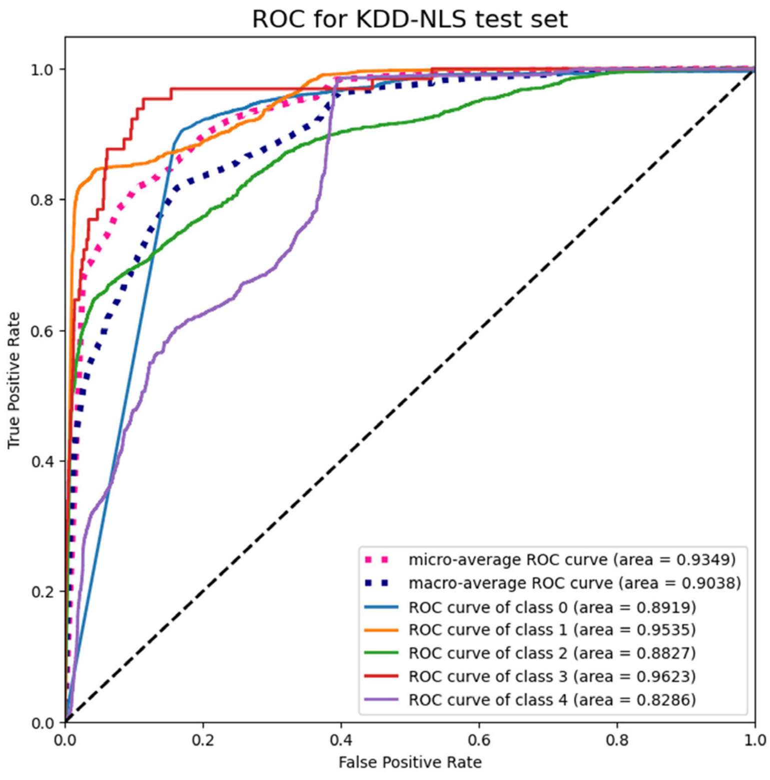
The LSTM-SMOTE model showed good results; however, the outcomes heavily depend on the amount of data and the training process. The learning process of large neural networks involves thousands of iterations and requires more layers of neurons.

In our case, LSTM is the classifier, and the model grows depending on the training data. The main errors, missed accidents, are not false alarms. The main reason is the deliberate training on test data. Typically, a neural network is not designed to be tested on data. This is a new solution to the problem. It is necessary to retrain the neural network with the data. To make the result more accurate, a 4-element classification was chosen. The graphs for 4-element classification are presented, showing the forecast before training and the results after training in Fig. 19.

Observations of the SMOTE algorithm's performance showed that the number of attacks in the data has significantly increased. SMOTE is designed to address data imbalance. However, in the main dataset, some accidents were not detected because there were too few or no such types, and SMOTE was unable to affect this. Fig. 20 presents the ROC curve.



a



b

Fig. 15. LSTM-SMOTE 4-item classification ROC curve: *a* — before training; *b* — after training

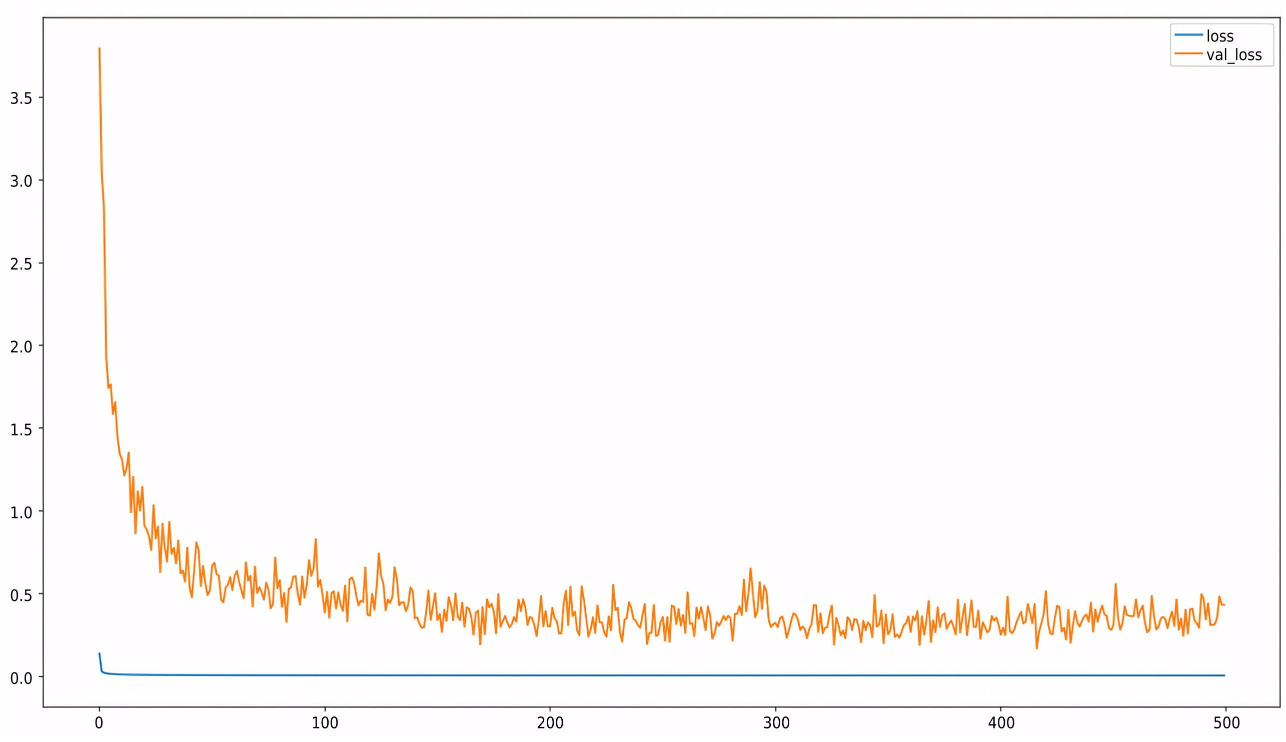
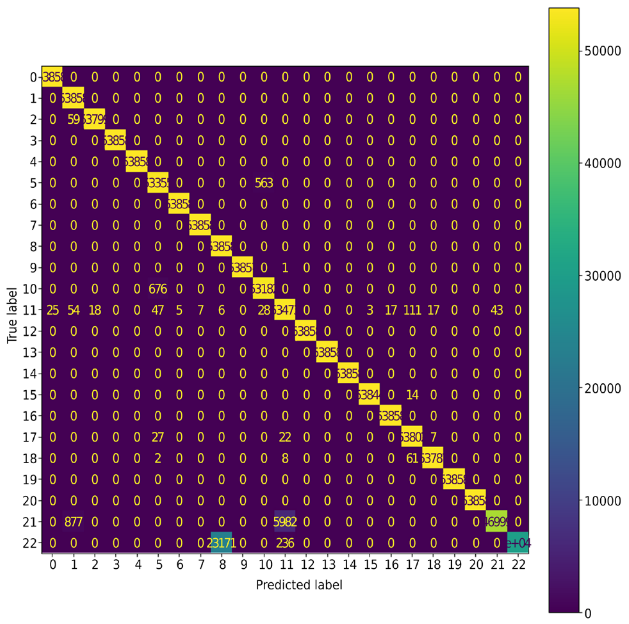
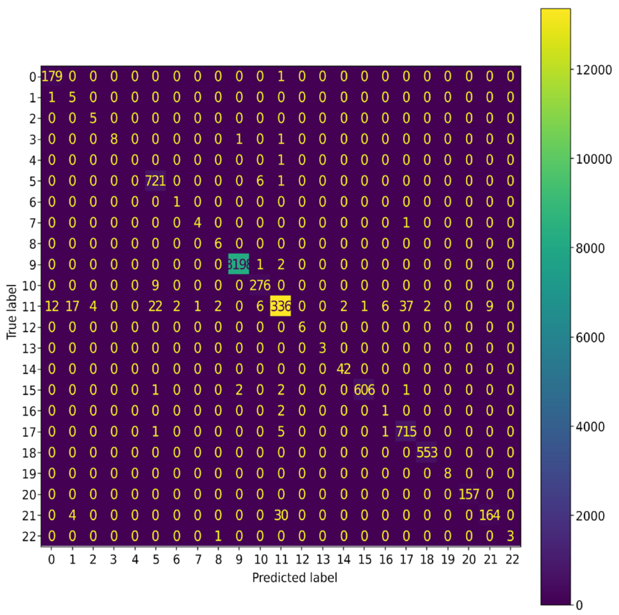


Fig. 16. Multiclass classification model



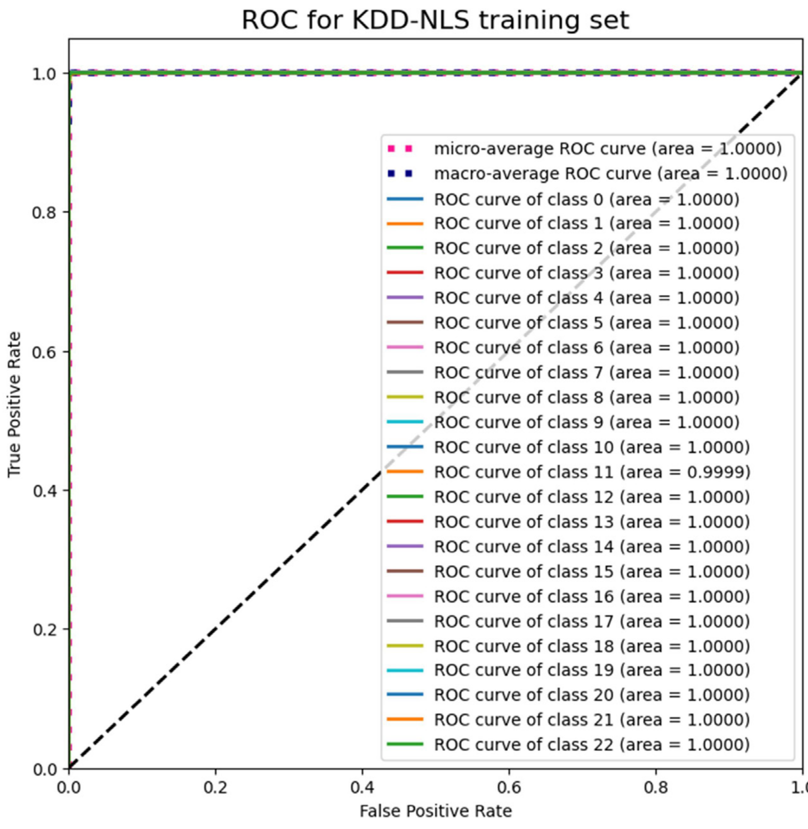
a



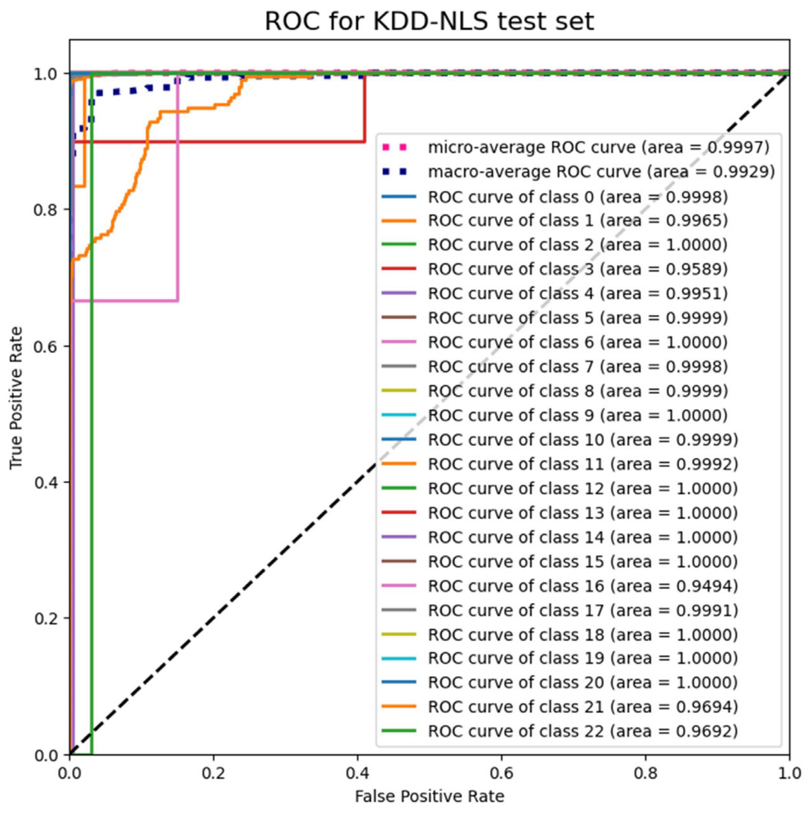
b

Fig. 17. Multi-classification True Positive, False Positive (TPFP):
a – train dataset;
b – test dataset

The binary and 4-element models will be the most indicative for us because our main task is to predict traffic anomalies. Determining the type of attack takes second place and is used as a complement to the first. Preparing data for multiclass classification is challenging because there may be very few examples of a specific type of attack. To summarize, the LSTM-SMOTE multiclass model's training, although highly dependent on its data, improves its prediction capability with each iteration.

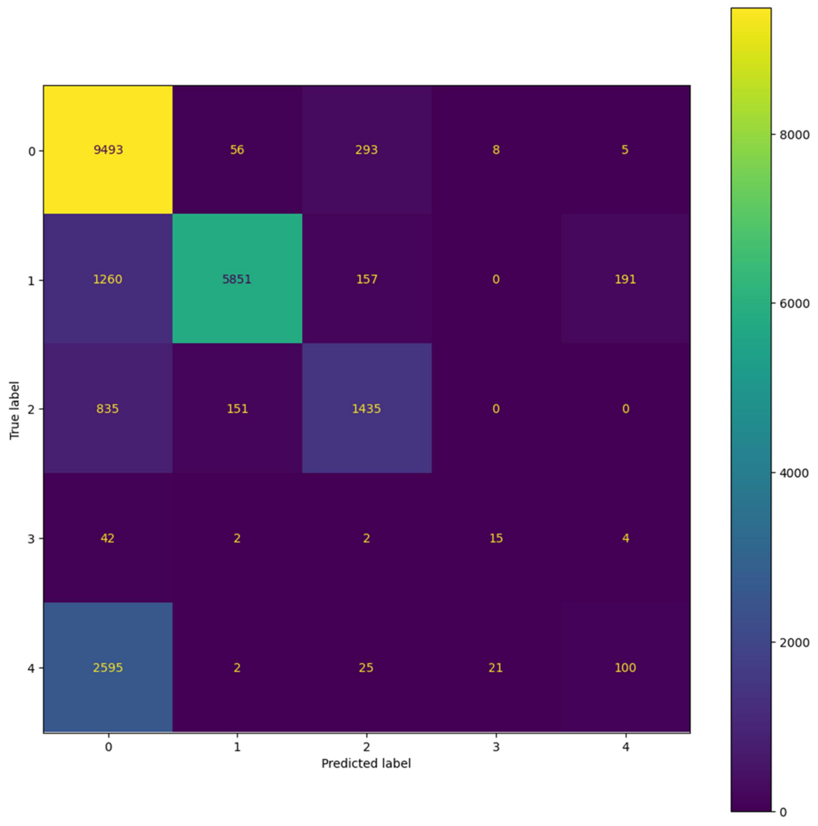


a

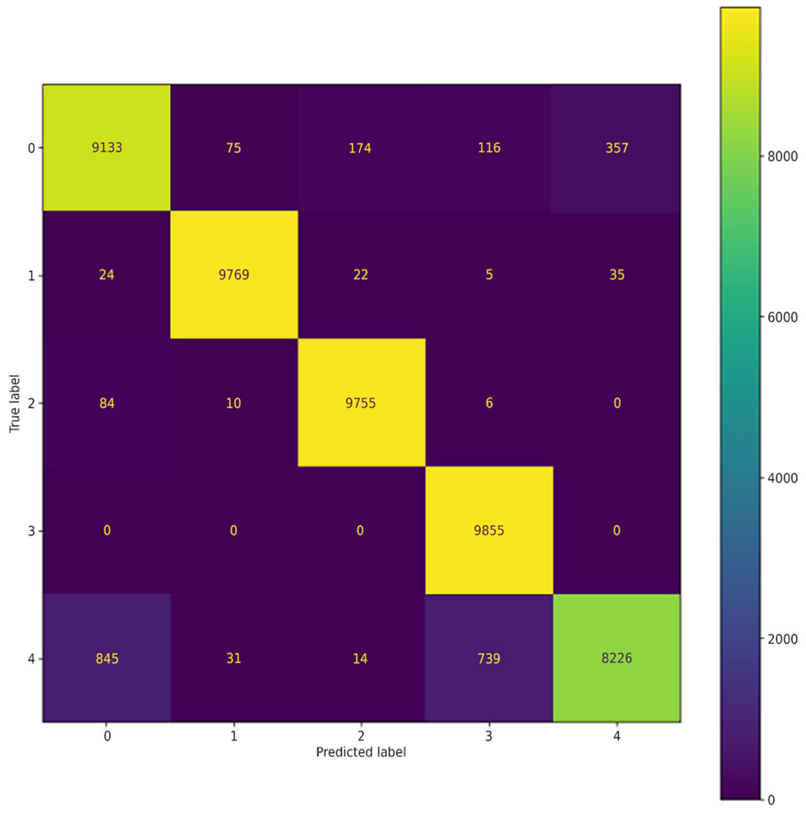


b

Fig. 18. ROC curve multi-classification: *a* – train; *b* – test

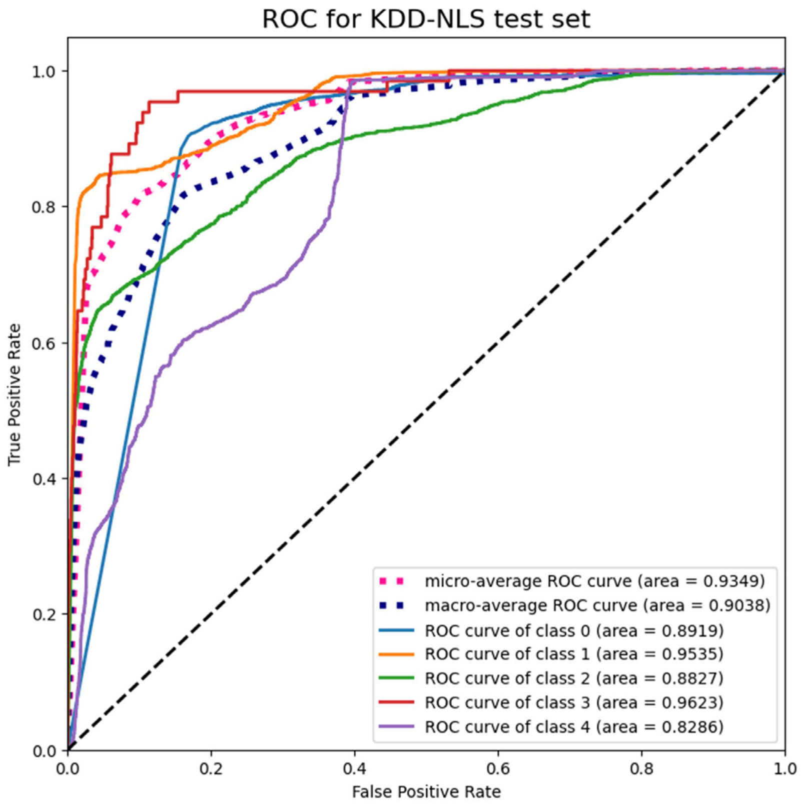


a

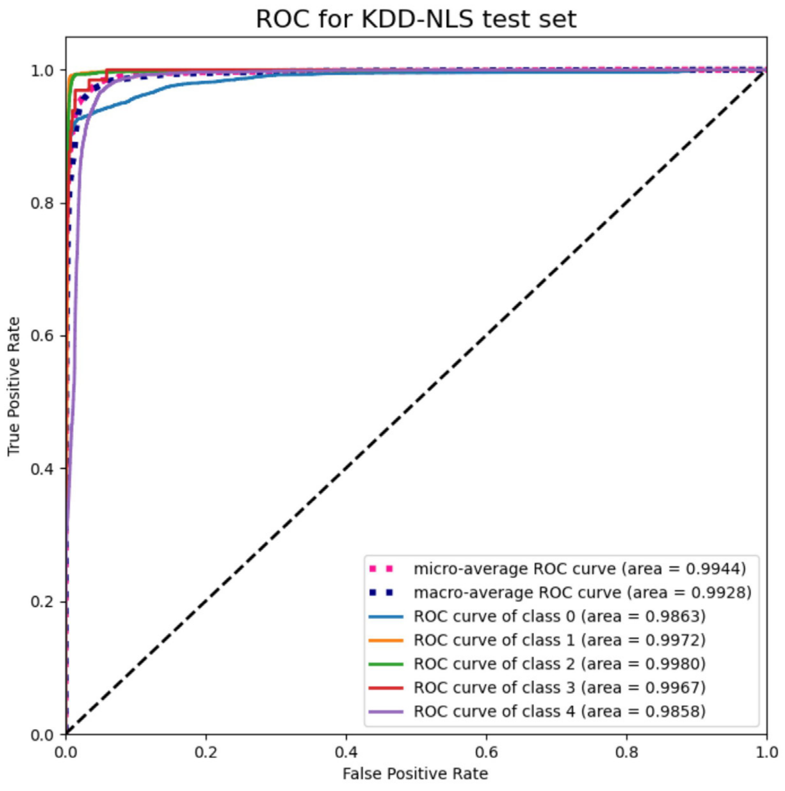


b

Fig. 19. Test: *a* – forecast before training; *b* – result after training



a



b

Fig. 20. ROC curve: *a* – before training; *b* – after training

6. Discussion of traffic anomaly prediction results using Long Short-term Memory (LSTM)

The use of a multi-class classification model (Fig. 1) has significantly improved the accuracy and granularity of detecting network attacks, which not only enhanced the overall prediction quality but also enabled a deeper threat diagnosis. Unlike binary classification, which divides traffic into two categories (normal/anomalous), multi-class classification allows for the identification of various types of attacks or malfunctions, making the system more flexible and informative.

As shown in Fig. 1, the accuracy of the multi-class classification model was lower than that of the binary model due to the increased complexity of separating network traffic into more than two categories. Nevertheless, using multi-class classification enabled more accurate identification of different types of threats (e.g., DDoS attacks, unauthorized access, and others). This, in turn, supports proactive attack response, allowing network operators to prepare in advance for potential threats instead of merely reacting to incidents that have already occurred.

The added classification granularity helps not only in anomaly detection but also in categorizing incidents by threat types, which can be used to create more accurate and context-aware protection methods. For example, upon detecting signs of a DDoS attack, the system can automatically trigger traffic-limiting measures, while in the case of identifying unauthorized access, it can activate security systems to investigate the incident.

Thus, the use of a multi-class classification model made it possible to significantly increase the level of detail in forecasting anomalies and network attacks, which could greatly improve the efficiency of cybersecurity systems.

The use of an LSTM model in Keras for forecasting network traffic and detecting anomalies (Fig. 2) has proven effective in identifying deviations from normal behavior. The model was configured to process network traffic time series and classify traffic states as either normal or anomalous. In the case of an anomaly, the system also determines the type of failure, allowing for more accurate threat diagnostics.

Fig. 2 presents the model architecture, including a folder with analysis code, the neural network itself, and result visualizations. This approach allows for a more detailed analysis of model performance, identifying weak spots and areas for improvement.

The results obtained using this model demonstrate that the system can effectively distinguish between normal and anomalous traffic, classifying attack types according to their characteristic patterns. Specifically, the main types of DDoS attacks identified are:

– probe – vulnerability scanning attacks;
– access – unauthorized access attempts;
– privilege – privilege escalation attacks.

Error handling and anomaly detection were also incorporated into the research framework, improving the system's resilience to data errors and enhancing its ability to self-learn on new data. As a result, upon detecting an anomaly, the system not only flags it but also accurately identifies the type of potential failure, which significantly improves monitoring and security processes.

Thus, the use of LSTM in this task not only improved anomaly detection but also enhanced the precision and timeliness of responses to various types of attacks, playing a critical role in real-world operation of telecommunications systems.

Within the study, binary and 4-class classification models proved to be the most effective in forecasting network traffic anomalies, as this task is the primary objective of the research. Unlike multi-class classification, the binary model focuses on detecting anomalous versus normal traffic, which is key for prompt response to network threats. The frequency and importance of real-time anomaly detection require high classification accuracy, making the binary model especially useful for practical applications.

The results presented in Fig. 9–12 confirm that the binary model provides high classification accuracy, minimizing task complexity and increasing system response speed. Analysis showed that this model strikes the best balance between sensitivity and specificity critical for effective threat detection and mitigation in real conditions.

A 4-class model was also used, which categorizes traffic into four groups. While this model is more complex, it adds an extra level of detail, useful in scenarios that demand more precise classification of attack types. For example, upon detecting an anomaly, the system can specify whether it relates to a DDoS attack, unauthorized access, privilege escalation, or another threat type. However, despite the added detail, the 4-class model shows slightly lower accuracy compared to the binary model due to increased classification complexity.

Preparing data for multi-class classification is a complex task. The challenge lies in the fact that examples of some attack types may be extremely rare, leading to a strong class imbalance. This makes model training more difficult and reduces its ability to effectively classify rare attack types, as shown in Fig. 7.

Despite the difficulties, the LSTM-SMOTE model for multi-class classification, thanks to the use of the Synthetic Minority Over-sampling Technique (SMOTE), demonstrated significant improvement with each training cycle. Each new training iteration enhanced the model's ability to identify rare attack types such as Privilege or Access, increasing its versatility and applicability in real-world conditions. Although multi-class classification requires more training time and tuning, its use significantly expands the system's capabilities for identifying threat types, providing more detailed information about ongoing incidents.

In conclusion, the binary model remains the most effective for quick forecasting of traffic anomalies, while the 4-class model offers a valuable supplement for detailed classification of attack types (Fig. 13–15). At the same time, despite the complexity of data preparation and class imbalance, the multi-class LSTM-SMOTE model shows considerable improvement with each training iteration and can be used for deeper analysis and more accurate forecasting of different types of attacks (Fig. 16–19), with the retraining results shown in Fig. 20.

The study has validated the effectiveness of neural network models – specifically the LSTM architecture combined with the SMOTE technique – for predicting anomalies in network traffic. The proposed approaches have shown strong performance in accurately and reliably detecting various types of attacks, including rare and less frequent ones.

At the same time, the suggested methods come with certain limitations that must be taken into account when deploying them in real-world environments. These include dependency on the quality and volume of input data, challenges in adapting the models to diverse network infrastructures, and limited interpretability of neural networks, which may complicate their integration into existing monitoring and security systems. Addressing these limitations will be essential when transitioning from research prototypes to practical implementations.

Looking ahead, the research can be further advanced by incorporating datasets from next-generation networks such as 5G and IoT, adopting explainable AI (XAI) approaches to improve model transparency, leveraging hybrid neural architectures, and enabling real-time stream processing of network data. Another promising area is the automation of model retraining for novel and previously unseen attack types, which could significantly enhance the resilience of anomaly detection systems against evolving cybersecurity threats.

7. Conclusions

1. A method for anomaly prediction in network traffic was implemented by developing three different neural network models:

– binary classification;

– 4-class classification;

– multiclass classification.

The binary model effectively distinguished between normal and abnormal traffic, categorizing attacks into DoS, Probe, Privilege, and Access types.

2. The development of the multiclass classification model encountered challenges due to the limited number of attack types. Although SMOTE was utilized to mitigate this issue, the prediction process remained imperfect when data was insufficient. Despite these challenges, the multiclass model was successfully developed, though its reliability was lower than that of the other models.

3. The primary limitation of the multiclass model was attributed to the train_test_split method, which evaluates performance based on training data but does not effectively measure generalization to new data. To address this, the model was retrained with additional data, and test data predictions were re-evaluated. This approach resulted in an accuracy exceeding 94 %. The LSTM-SMOTE multi-classification model demonstrated the highest performance, with predictive accuracy improving from 75 % to 99 % across iterations, highlighting its strong dependence on data quality and quantity.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has associated data in a data repository.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Cheng, Y., Liu, Q., Wang, J., Wan, S., Umer, T. (2018). Distributed Fault Detection for Wireless Sensor Networks Based on Support Vector Regression. Wireless Communications and Mobile Computing, 2018 (1). https://doi.org/10.1155/2018/4349795
2. Muriira, L. M., Zhao, Z., Min, G. (2018). Exploiting Linear Support Vector Machine for Correlation-Based High Dimensional Data Classification in Wireless Sensor Networks. Sensors, 18 (9), 2840. https://doi.org/10.3390/s18092840
3. Latif, Z., Umer, Q., Lee, C., Sharif, K., Li, F., Biswas, S. (2022). A Machine Learning-Based Anomaly Prediction Service for Software-Defined Networks. Sensors, 22 (21), 8434. https://doi.org/10.3390/s22218434
4. Song, W., Beshley, M., Przystupa, K., Beshley, H., Kochan, O., Pryslupskyi, A. et al. (2020). A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection. Sensors, 20 (6), 1637. https://doi.org/10.3390/s20061637
5. Oliveira, T. P., Barbar, J. S., Soares, A. S. (2016). Computer network traffic prediction: a comparison between traditional and deep learning neural networks. International Journal of Big Data Intelligence, 3 (1), 28. https://doi.org/10.1504/ijbdi.2016.073903
6. Alkasassbeh, M. (2018). A Novel Hybrid Method for Network Anomaly Detection Based on Traffic Prediction and Change Point Detection. Journal of Computer Science, 14 (2), 153–162. https://doi.org/10.3844/jcssp.2018.153.162
7. Khan, I. A., Pi, D., Khan, Z. U., Hussain, Y., Nawaz, A. (2019). HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems. IEEE Access, 7, 89507–89521. https://doi.org/10.1109/access.2019.2925838
8. Bhatia, R., Benno, S., Esteban, J., Lakshman, T. V., Grogan, J. (2019). Unsupervised machine learning for network-centric anomaly detection in IoT. Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, 42–48. https://doi.org/10.1145/3359992.3366641
9. Nurzhaubayeva, G., Haris, N., Chezhimbayeva, K. (2024). Design of the Wearable Microstrip Yagi-Uda Antenna for IoT Applications. International Journal on Communications Antenna and Propagation (IRECAP), 14 (1), 24. https://doi.org/10.15866/irecap.v14i1.24315
10. Chezhimbayeva, K., Konyrova, M., Kumyzbayeva, S., Kadylbekkyzy, E. (2021). Quality assessment of the contact center while implementation the IP IVR system by using teletraffic theory. Eastern-European Journal of Enterprise Technologies, 6 (3 (114)), 64–71. https://doi.org/10.15587/1729-4061.2021.244976
11. Mukhamejanova, A. D., Grabs, E. A., Tumanbayeva, K. K., Lechshinskaya, E. M. (2022). Traffic simulation in the LoRaWAN network. Bulletin of Electrical Engineering and Informatics, 11 (2), 1117–1125. https://doi.org/10.11591/eei.v11i2.3484