УДК 004.032.26, 620.92 https://doi.org/10.36296/1819-8058.2024.2(77)52-57

DETERMINATION OF NETWORK TRAFFIC ANOMALIES IN A DISTRIBUTED COMPUTER SYSTEM WITH ENERGY FACILITIES

Received Jun. 03, 2024; accepted Jun. 21, 2024
Available online Jul. 01, 2024

Shapovalova S.[1], Matiakh S.[2], Titov V.[3]

Author for correspondence: Matiakh Serhii,
e-mail: krypto@ukr.net

[1] PhD, Assoc. Prof.
https://orcid.org/0000-0002-3431-5639

[2] PhD
https://orcid.org/0000-0002-1707-3519

[3] M. S.
https://orcid.org/0009-0000-5780-5596

[1,2,3] National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine

[2] Institute of Renewable Energy, National Academy of Science of Ukraine, Kyiv, Ukraine

Abstract. *The paper presents research, the purpose of which is to define a machine learning model for express analysis of network traffic in a distributed computer system for managing decentralized generation facilities based on renewable energy. The available tools for monitoring processes in the computer network are considered. The problem of network traffic anomaly detection is presented as a binary classification problem. The input data of the model is represented by 10 features, which are determined on the basis of the RFE method based on the results of computational experiments on classification based on the examples of the Network Intrusion Detection dataset. To determine the optimal model, both neural network models: MLP, RNN, LSTM, and traditional machine learning models: KNN, Logistic Regression, Decision Tree, GBM were investigated. Implementations of these models from Scikit-learn and TensorFlow resources were configured and trained. According to the results of computational experiments, the most accurate models for detecting network traffic anomalies in a distributed computer system were determined: LSTM (F1 score: 0.97 on the test sample, 0.95 - in working mode) and RNN (F1 score: 0.97 on the test sample, 0.94 - in working mode). Tests have shown that RNN or LSTM anomaly prediction when transmitting a packet does not change the time order relative to transmission without prediction. The use of defined machine learning models to build subsystems for detecting network traffic anomalies in systems of distributed generation of electric energy based on renewable sources will allow to increase resistance to failures and force majeure situations, the efficiency of their work, especially taking into account the challenges to the operation of the energy system of Ukraine during the war.*

Keywords: *renewable energy, energy decentralization, MLP, RNN, LSTM, KNN, Logistic Regression, Decision Tree, GBM.*

ВИЗНАЧЕННЯ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ В РОЗПОДІЛЕНІЙ КОМП'ЮТЕРНІЙ СИСТЕМІ З ЕНЕРГЕТИЧНИМИ ОБ'ЄКТАМИ

Отримано 03 чер. 2024 р.; рекомендовано до публікації 21 чер. 2024 р.
Доступно онлайн 01 лип. 2024 р.

Шаповалова С. І.[1], Матях С. В.[2], Тітов В. М.[3]

Автор для кореспонденції: Матях Сергій,
e-mail: krypto@ukr.net

[1] канд. техн. наук, доцент
https://orcid.org/0000-0002-3431-5639

[2] канд. техн. наук
https://orcid.org/0000-0002-1707-3519

[3] магістрант
https://orcid.org/0009-0000-5780-5596

[1, 2, 3] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

[2] Інститут відновлюваної енергетики НАН України, м. Київ, Україна

Анотація. *У роботі описано дослідження, метою яких є визначення моделі машинного навчання для експрес-аналізу мережевого трафіку в розподіленій комп'ютерній системі управління об'єктами децентралізованої генерації на основі відновлюваної енергетики. Розглянуто наявні засоби моніторингу процесів у комп'ютерній мережі. Представлено задачу виявлення аномалій мережевого трафіку як задачу*

бінарної класифікації. Вхідні дані моделі представлено 10 ознаками, які визначені на основі методу RFE за результатами проведених обчислювальних експериментів з класифікації за прикладами датасету Network Intrusion Detection. Для визначення оптимальної моделі було досліджено як нейромережеві моделі MLP, RNN, LSTM, так і традиційні моделі машинного навчання KNN, Logistic Regression, Decision Tree, GBM. Налаштовано та проведено навчання реалізацій цих моделей з ресурсів Scikit-learn, TensorFlow. За результатами обчислювальних експериментів визначено оптимальні за точністю моделі для виявлення аномалій мережевого трафіку в розподіленій комп'ютерних системах LSTM (F1 score: 0,97 на тестовій виборці, 0,95 – у робочому режимі) та RNN (F1 score: 0,97 на тестовій виборці, 0,94 – у робочому режимі). Випробовування показали, що передбачення аномалій RNN або LSTM при передачі пакету не змінює порядок часу відносно передачі без передбачення. Використання визначених моделей машинного навчання для побудови підсистем виявлення аномалій мережевого трафіку в системах розподіленої генерації електричної енергії на основі відновлюваних джерел дозволить підвищити стійкість до відмов та форс-мажорних ситуацій, ефективність їх роботи, особливо враховуючи виклики до роботи енергетичної системи України під час війни.

Ключові слова: *відновлювана енергетика, децентралізації енергетики, MLP, RNN, LSTM, KNN, Logistic Regression, Decision Tree, GBM.*

List of abbreviations and acronyms used:

CNN – Convolutional Neural Network

DNN – Deep Neural Networks

GBM – Gradieent boosting machine

KNN – k-Nearest Neighbors

LSTM – Long Short-Term Memory

MLP – Multilayer Perceptron

RFE – Recursive Feature Elimination)

RNN – Recurrent Neural Network

Introduction

The Energy Strategy of Ukraine until 2050, approved by the Cabinet of Ministers of Ukraine on April 21, 2023, envisages increasing the sustainability of the Ukrainian energy system and the reliability of energy supply by decentralizing electricity generation throughout the country, which is significantly related to the development of renewable energy. Amendments to the Laws of Ukraine on Restoration and Green Transformation of the Energy System of Ukraine [1] are aimed at reforming the electricity industry in many aspects of the functioning and development of the Ukrainian energy market, including small distributed generation. In addition to their economic benefits, distributed power generation systems can perform a balancing function. The transformation of the energy system implies a significant complication of the distributed power system and requires the creation of effective monitoring systems to ensure the stability and reliability of energy supply [2]. Subsystems of power facilities are distributed computer systems. Such systems use heterogeneous software that must exchange information. One aspect of security is the timely detection of errors in the results of the system components. Such errors may be related to flaws inherent in the software development – the so-called "bugs" – or caused by external factors directly during the execution of programs in the complex. Even minor errors in the operation of programs can lead to serious consequences, such as reduced performance, data loss, or a computer network security breach. This is critically important for distributed computer systems that contain energy facilities. To detect program execution errors in a timely manner, it is necessary to perform

- continuous monitoring of running programs and processes;
- control of operating system resources;
- control of the components' connection health;
- component load control;
- monitoring of network traffic.

To solve the first four tasks, special software tools are used, the most common of which are: Nagios, Zabbix, and Prometheus. These tools are effective for basic monitoring and control, however, they are not always suitable for networks with energy facilities due to

- Limitations of integration with other monitoring and control tools;

- scalability issues (as the size of the infrastructure increases, the performance of these tools can significantly decrease);

- the inability to ensure continuous monitoring of certain parameters in real time, which can lead to delays in detecting threatening situations.

To solve the fifth task of network traffic analysis, special commercial software is also used, which is multifunctional, requires a lot of computing resources, and usually has high requirements for computing resources, as well as for the knowledge and experience of personnel. Table 1 provides information on the most common network traffic monitoring tools, as well as an assessment of the effectiveness of their analysis according to the PAT Researcher platform editor [3].

Table 1. Network traffic monitoring tools

Name	Integration capabilities	Analysis methods	Evaluation of analysis results
Splunk	API для Python, JavaScript	Machine learning methods (uses Splunk Machine Learning Toolkit)	8.2
Crunch Metrics	API for Java, Python; API for integration with other services such as AWS, Google Cloud	Methods of statistical analysis	7.6
Elastic X-Pack	API for Java; integration with Elasticsearch	Self-learning algorithms, neural network models	9.1
Loom Systems	API for Python, JavaScript	AI for behavioral analysis and cyber threat detection	9.5
Anodot	API for Java, Python; APIs for integration with other services such as AWS, Azure, Google Cloud	Machine learning methods, correlation analysis	8.9 (real-time analysis)
Numenta	API for Python, C++	Data analysis methods NetFlow, J-Flow, sFlow, IPFIX; neural network models (HTM - Hierarchical Temporal Memory).	8.5 (real-time analysis)
DataRPM	API for Java, Python, integration with Hadoop, Spark, Cassandra services	Machine learning methods, including deep learning	7.5

The article [4] evaluated the basic concepts of artificial intelligence for detecting anomalies in a computer network, which were interpreted as changes in the regular behavior of the network. The tests were conducted using examples from the Intrusion Detection Evaluation Dataset (UNB ISCX IDS 2012) [5]. Based on the evaluation results, a model based on Random Forest was selected for implementation. The results of DNN (Deep Neural Networks) were close in accuracy. The third alternative was a two-level Naïve Bayes model. The article identifies the advantages of using machine learning methods to detect anomalies in the network, in particular: close to real-time detection, which is necessary for effective network security; sufficiently high accuracy when dealing with complex and dynamic network traffic and the ability to adapt to network environments.
In [6], a concept based on a combination of a convolutional neural network CNN (Convolutional Neural Network) and a recurrent network LSTM (Long Short-Term Memory) with attention mechanisms was proposed for analyzing network traffic. The resulting model has the ability to successfully adapt to different data sets and achieve high accuracy rates. The model can also work effectively with encrypted traffic, making it a valuable tool for security and network management applications. The experimental results presented in this paper demonstrate the high accuracy of the model in classification tasks. On a dataset of real VPN-nonVPN [7] traffic, the model showed an F1 accuracy of over 95% and a recall of over 90%.

The above works and the software tools used in this paper involve multitasking analysis, and the models created require a lot of computing resources. Distributed computer networks with power generation sources can be of different scales, which primarily concerns the computing power involved. Precisely for renewable energy, in connection with the need to integrate into the energy supply system elements related to different types of renewable sources and managed by different network systems [8], it is relevant and practically significant to provide express analysis of network traffic based on modern machine learning models with using the minimum number of computing resources.

Statement of the problem

The object of research is software tools for detecting program execution errors in a distributed computer system.

The subject of research is artificial intelligence models for analyzing network traffic in a distributed computer control system.

The purpose of the study is to determine a machine learning model for rapid analysis of network traffic in a distributed computer system for managing decentralized renewable energy facilities.

To achieve this goal, the following tasks are necessary:

1. Formalize the problem for the artificial intelligence model.
2. Identify machine learning models for detecting network traffic anomalies.
3. Determine the parameters and hyperparameters of the models and train them.
4. To conduct computational experiments to determine the optimal model for detecting network traffic anomalies in a distributed computer system in terms of accuracy and speed.

Research methods: machine learning models: KNN, Logistic Regression, Decision Tree, GBM; neural network models: MLP, RNN, LSTM.

1. The task for the artificial intelligence model.

For rapid analysis, it is proposed to solve the problem of binary classification, the result of which is a prediction of the presence of an anomaly: 1 (present), 0 (perceptible), or probability of presence/absence, depending on the model.

Compared to n-ary classification (n>2), binary classification simplifies the model, which contributes to its effective training, reduces the risk of overtraining in the case of neural networks, speeds up predictions, and requires less computing resources.

The model must learn the ability to effectively distinguish between normal and abnormal network traffic.

To determine the input features of the binary classification task, we used the Network Intrusion Detection dataset from the Kaggle platform [9]. Examples of this dataset have 42 characteristics each, determined by the values of the technical parameters of packets and network traffic or calculated on their basis.

According to the results of computational experiments on all these models, using the RFE (Recursive Feature Elimination) method, the 10 most significant features for predicting the presence of anomalies were identified. Subsequently, the Network Intrusion Detection dataset was transformed into a derivative dataset with a smaller input vector dimension.

Thus, the input data for detecting network traffic anomalies are as follows:

1. ProtocolType – protocol type (for example, TCP, UDP, ICMP).
2. Flag – the state of the TCP connection flags.
3. SrcBytes – the number of bytes sent from the source to the destination.
4. DstBytes – the number of bytes sent from the recipient to the source.
5. Count – the number of connections for the last 2 seconds for the same host.
6. SameSrvRate – the share of connections for the same service.
7. DiffSrvRate – the share of connections for different services.
8. DstHostSrvCount – the number of connections to the same service of the receiving host for the last 2 seconds.
9. DstHostSameSrvRate – the proportion of connections to the same service of the receiving host.
10. DstHostSameSrcPortRate – the share of connections from the same source port to the destination host.

2. Machine learning models for detecting network traffic anomalies.

Several artificial intelligence concepts can be used to solve this problem. To determine the optimal model, we investigated neural network models: MLP, RNN, LSTM, and traditional machine learning models: KNN, Logistic Regression, Decision Tree, GBM.

When choosing models, we took into account the peculiarities of the classification task, first of all, the importance of taking into account information on the analysis of previous network traffic packets. For this purpose, the most suitable are recurrent neural networks RNN and their variant – LSTM [10]. Other models have been investigated as traditional classification models. Some of these models can be applied to other tasks with time-dependent input data. For example, in [11], a comparative analysis of KNN and Light GBM Algorithms was conducted for the task of wind energy forecasting. In [12], the examples of the dataset were measurement data for 7 parameters made over a 17-hour period, and out of 12 models, Randomized Decision Trees was determined to be the best in terms of accuracy for the task of binary classification of the current state of stamping presses.

3. Implementation of machine learning models

Based on the RFE method, the weights of each input feature for all models were determined (Table 2).

Table 2. Weights of input parameters

Input feature	KNN	Logistic regression	Decision Tree	GBM	LSTM	RNN	MLP
ProtocolType	0.14	0.14	0.25	0.04	0.05	0.03	0.15
Flag	0.07	0.08	0.01	0.06	0.03	0.21	0.14
SrcBytes	0.35	0.30	0.33	0.71	0.35	0.28	0.252
DstBytes	0.25	0.19	0.19	0.07	0.34	0.27	0.23
Count	0.06	0.06	0.035	0.015	0.012	0.012	0.02
SameSrvRate	0.01	0.05	0.05	0.012	0.014	0.014	0.014
DiffSrvRate	0.03	0.05	0.03	0.02	0.012	0.01	0.011
DstHostSrvCount	0.03	0.05	0.042	0.035	0.1	0.1	0.06
DstHostSameSrvRate	0.028	0.03	0.03	0.015	0.012	0.007	0.054
DstHostSameSrcPortRate	0.032	0.05	0.033	0.023	0.08	0.067	0.069

Table 3 shows the tools for creating machine learning models from the Scikit-learn [13] and TensorFlow [14] resources.

Table 3. Implementation of machine learning models

Model name	Implementation tools
KNN	Tools Scikit-learn, library sklearn.neighbors, class KNeighborsClassifier
Logistic Regression	Tools Scikit-learn, library sklearn.linear_model, class LogisticRegression
Decision Tree	Tools Scikit-learn, library sklearn.tree, class DecisionTreeClassifier
GBM	Tools Scikit-learn, library sklearn.ensemble, class GradientBoostingClassifier
LSTM	Tools tf2onnx [15], tensorflow, library tf.keras.layers, class LSTM
RNN	Tools tensorflow, library tf.keras.layers, class SimpleRNN
MLP	Tools scikit-learn, library sklearn.neural_network, class MLPClassifier

4. Computational experiments

All models were trained on the Network Intrusion Detection dataset. The training sample was 80% of the dataset, and the test sample was 20%.

The first set of experiments was conducted to determine the optimal model in terms of classification accuracy.

For the experiments, in addition to the test sample of the Network Intrusion Detection dataset, a set of real data was created, determined by the current characteristics of network traffic in a distributed computer network. Based on the metrics, 17885 examples were created.

The classification accuracy for each model was evaluated separately on the test set and the data set obtained in the production mode. The results of the F1 score are shown in Table 4.

Table 4. Classification results by accuracy

Model name	Accuracy by F1 score		
	on the training dataset	on the test dataset	in the working mode
KNN	0.93	0.90	0.81
Logistic Regression	0.92	0.91	0.71
Decision Tree	0.97	0.97	0.64
GBM	0.93	0.93	0.88
LSTM	0.98	**0.97**	**0.95**
RNN	0.97	**0.97**	**0.94**
MLP	0.94	0.93	0.61

The difference between the classification accuracy on test and real data is explained by the fact that: 1) models can be retrained on training data, which leads to a loss of generalizability when working with new data; 2) test datasets usually have a similar structure and distribution to training data, while real data may contain more noise, irregularities, and unknown variants, which affects the accuracy of models.

The best results in terms of accuracy on the test set and real data were obtained by LSTM and RNN neural networks. The worst result was obtained by MLP.

The second set of experiments was conducted to determine the optimal model in terms of classification time.

The experiment consisted in determining the average transmission time of 1 packet without connecting the artificial intelligence model and with the connection for anomaly detection. The packet was transmitted 1000 times using each method. After that, the average transmission time was determined. The following tools were used for the experiment: onnxruntime [16] – to connect trained models to the system; sharppcap [17] – to listen to the main network device; Stopwatch from the System.Diagnostics library [18] – to determine the data transfer time.

Based on the experimental results, we determined the approximate classification time for each of the three models that provide the most accurate predictions (Table 5).

Table 5. Average classification time when transferring 1 data packet

Model name	Average data transmission time, s		Classification time, s
	without classification	with classification	
GBM	0.18	0.23	0.05
RNN	0.2	0.47	0.27
LSTM	0.19	0.53	0.34

The RNN and LSTM neural network models spend more time on prediction. However, the order of the numbers that determine the transmission time with and without prediction of these models is the same.

Conclusions

1. Based on the results of computational experiments using examples of the Network Intrusion Detection dataset based on the RFE method, the 10 most significant features were identified and used as input data for the task of binary classification of network traffic anomalies.
2. Predictive models were determined: KNN, Logistic Regression, Decision Tree, GBM, MLP, RNN, LSTM.
3. We have configured and trained implementations of these models from the resources Scikit-learn, TensorFlow.
4. According to the results of computational experiments, the models that are optimal in terms of accuracy for detecting network traffic anomalies in a distributed computer system are determined: LSTM (F1 score: 0.97 on the test sample, 0.95 in the working mode) and RNN (F1 score: 0.97 on the test sample, 0.94 in the working mode).

Thus, the use of defined machine learning models for the construction of network traffic anomaly detection subsystems in distributed electrical generation systems based on renewable sources will allow to increase their reliability and stability, especially considering the instability of the energy system of Ukraine during the war.

REFERENCES

1. Про внесення змін до деяких законів України щодо відновлення та "зеленої" трансформації енергетичної системи України/ Закон України № 3220-IX. ВВР. 2023. № 82, с. 301. https://zakon.rada.gov.ua/laws/show/3220-IX#Text
2. D. Bondarenko, S. Matyakh, T. Surzhyk, I. Sheiko Aspects of the further development of photoenergy according to the materials of the scientific and practical conference «Renewable energy and energy efficiency in the 21st century» 2023. Vidnovljuvana energetyka. (2023), No. 4, 39-44 (in Ukrainian). https://doi.org/10.36296/1819-8058.2023.4(75).39-44
3. TOP 10 anomaly detection software, PAT RESEARCH, Available at: https://www.predictiveanalyticstoday.com/top-anomaly-detection-software/. last accessed 03.06.2024).
4. Estévez-Pereira J.J., Fernández D., Novoa F.J. Network anomaly detection using machine learning techniques. Proceedings of 3rd XoveTIC Conference, 2020, 54(1), 8. https://doi.org/10.3390/proceedings2020054008
5. Intrusion detection evaluation dataset (ISCXIDS2012). University of New Brunswick. Available at: https://www.unb.ca/cic/datasets/ids.html (last accessed 03.06.2024).
6. Hu F., Zhang S., Lin X. et al. Network traffic classification model based on attention mechanism and spatiotemporal features. EURASIP J. on Info. Security. 2023, 6. https://doi.org/10.1186/s13635-023-00141-4
7. VPN-nonVPN dataset (ISCXVPN2016). University of New Brunswick. Available at: https://www.unb.ca/cic/datasets/vpn.html (last accessed 03.06.2024).
8. D. Bondarenko, S. Matyakh, T. Surzhyk, V. Shevchuk Energy unit kit for photovoltaic cluster. Vidnovljuvana energetyka. (2023), No.3, 53-58 (in Ukrainian). https://doi.org/10.36296/1819-8058.2023.3(74).53-58
9. Network Intrusion Detection. Kaggle/Competitions. Available at: https://www.kaggle.com/code/ahmedsaed26/99-6-accuracy-network-intrusion-detection/input (last accessed 03.06.2024).
10. Malhotra P., Vig L., Shroff G., Agarwal P. Long Short Term Memory Networks for Anomaly Detection in Time Series. ESANN 2015 proceedings. Available at: https://www.researchgate.net/publication/304782562_Long_Short_Term_Memory_Networks_for_Anomaly_Detection_in_Time_Series (last accessed 03.06.2024).
11. Sushree S. P., Ashwin K. S., Rajesh P. A Comparative Analysis of KNN and Light GBM Algorithms for Wind Energy Forecasting. CCPIS. 2023, pp. 1-4. doi: 10.1007/978-3-031-45630-5_7.
12. Coelho D., Costa D., Rocha E. M., Almeida D., Coelho S. D. Predictive maintenance on sensorized stamping presses by time series segmentation, anomaly detection, and classification algorithm. Procedia Computer Sciencею. Vol. 200. 2022. P. 1184-1193. https://doi.org/10.1016/j.procs.2022.01.318.
13. Pedregosa et al. Scikit-learn: Machine Learning in Python. JMLR 12, 2011, pp. 2825-2830.
14. Abadi M., Agarwal A., Barha P. et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Available at: https://www.tensorflow.org/ (last accessed 03.06.2024).
15. tf2onnx - Convert TensorFlow, Keras, Tensorflow.js and Tflite models to ONNX. Available at: https://github.com/onnx/tensorflow-onnx (last accessed 03.06.2024).
16. Tutorial: Detects objects using ONNX in ML.NET. Available at: https://learn.microsoft.com/en-us/dotnet/machine-learning/tutorials/object-detection-onnx (last accessed 03.06.2024).
17. The official SharpPcap repository. Available at: https://github.com/dotpcap/sharppcap (last accessed 03.06.2024).
18. System.Diagnostics Namespace. Available at: https://learn.microsoft.com/en-us/dotnet/api/system.diagnostics?view=net-8.0 (last accessed 03.06.2024)