energies

MDPI

Article

Internet Threat Detection in Smart Grids Based on Network Traffic Analysis Using LSTM, IF, and SVM

Szymon Stryczek † and **Marek Natkaniec** *,†

Institute of Telecommunications, AGH University of Science and Technology, Mickiewicza 30, 30-059 Krakow, Poland
* Correspondence: natkanie@agh.edu.pl
† These authors contributed equally to this work.

Abstract: The protection of users of ICT networks, including smart grids, is a challenge whose importance is constantly growing. Internet of Things (IoT) or Internet of Energy (IoE) devices, as well as network resources, store more and more information about users. Large institutions use extensive security systems requiring large and expensive resources. For smart grid users, this becomes difficult. Efficient methods are needed to take advantage of limited sets of traffic features. In this paper, machine learning techniques to verify network events for recognition of Internet threats were analyzed, intentionally using a limited number of parameters. The authors considered three machine learning techniques: Long Short-Term Memory, Isolation Forest, and Support Vector Machine. The analysis is based on two datasets. In the paper, the data preparation process is also described. Eight series of results were collected and compared with other studies. The results showed significant differences between the techniques, the size of the datasets, and the balance of the datasets. We also showed that a more accurate classification could be achieved by increasing the number of analyzed features. Unfortunately, each increase in the number of elements requires more extensive analysis. The work ends with a description of the steps that can be taken in the future to improve the operation of the models and enable the implementation of the described methods of analysis in practice.

Keywords: smart grids; traffic analysis; threat detection; limited set of features; machine learning

Citation: Stryczek, S.; Natkaniec, M. Internet Threat Detection in Smart Grids Based on Network Traffic Analysis Using LSTM, IF, and SVM. *Energies* **2023**, *16*, 329. https://doi.org/10.3390/en16010329

Academic Editor: Álvaro Gutiérrez

Received: 9 December 2022
Revised: 19 December 2022
Accepted: 21 December 2022
Published: 28 December 2022

1. Introduction

A modern smart grid is characterized by the possibility of unexpected events occurring in it. Their proper identification is the key to ensuring user safety. Only some users have the skills to identify online threats based on their characteristics. Therefore, it is essential to automate this process as much as possible. This is where machine learning techniques can help. They allow for the identification of threats and abnormal events in a way that does not require human intervention. The resources and methods used for such analysis are a constant challenge. An additional aspect is the issue of model training. The training sets must be universal enough for the models to be able to indicate the threat after the training process unambiguously. Operating on large volumes of data simplifies searching for anomalies in data traffic. However, not every network anomaly phenomenon is a threat. Searching for events that deviate from the standards is only the first step; then, it should be determined whether the anomalies under investigation pose a threat. The problem is serious and requires decisive action. According to a SonicWALL annual report, only in 2021, the number of ransomware attacks and encrypted threats increased by 105% and 167%, respectively [1]. This is mainly a problem of large ICT companies on the market, where extensive security methods and complex models can be used. Collecting and examining many parameters helps detect threats but is very resource-intensive. It also comes with expenses. Smart grid users and small organizations usually do not have access to huge computing power. It is important to simplify the methods in such a way as to minimize the

loss of efficiency while limiting the required resources. According to the reports from the same organization, the number of IoT malware increased between 2019 and 2021 by 50%, 66%, and 6%, year by year [1–3]. The problem is growing and requires appropriate action. Enormous increases in registered incidents have been observed in recent years. The report reveals a significant growth in malware threats. Malware attacks are dangerous for all users. They can also be completely invisible to them. Network traffic analysis is a method that allows for the detection of such activities and protects end customers against attacks.

Many types of cyber-attacks threaten energy infrastructure. From 2010 to July 2022, sixteen severe worldwide attacks on grids were identified [4]. One each was registered in Africa and Southern America, three in Northern America, four in Asia, and as many as seven in Europe. The last mentioned threat was targeted at an Energy Company in Ukraine in April 2022 [5]. Attackers used malware called Industroyer2 which can control power flows in a grid. Attacks on energy infrastructure attract public attention because such attacks can affect the community's life. In the mentioned above article [4], the authors indicated that the development of modern energy infrastructure, especially smart grids needs a secure communication infrastructure with advanced technologies such as artificial intelligence or blockchain. One of the most important aspects is the transformation of grids by making energy infrastructure more autonomous. One of the solutions to reach that goal is to come into use Internet of Energy (IoE) devices. IoE is based on the same principles as the Internet of Things (IoT). Moreover, similar to IoT, one critical issue is offering privacy and secure connections for users of IoE devices. In [6], apart from proposals for the use of IoE solutions to reduce the environmental impact of the energy production process, the importance of data integrity and confidentiality in IoE applications is also emphasized. Confidentiality of energy consumption information in an institution can be crucial for proper functioning. In every place where small IoE devices with limited computing power would be used, it is necessary to implement threat detection methods that optimize the use of available resources.

The threat can be characterized by traffic source, target, used port, or protocol. When a large amount of traffic is exchanged, Access Control Lists (ACLs) on the network device can be added. These lists allow for traffic filtering with defined characteristics and to specify whether the lists should be used for incoming or outgoing traffic. For well-known sources of unsafe traffic, the concept of denylist can be used. Denylists contain a set of entries that identify the source of traffic and usually block all traffic coming from it. This is a solution used, for example, in the e-mail service to block unwanted messages. All the examples described above represent safeguards against known threats, and for such protection to be set, it is necessary to detect the danger and describe it unambiguously. Thus, the system effectively protects users against already-known threats. In the event of a new threat, using the above methods, the victim becomes defenseless, and if he does not observe anything disturbing, the list of victims will continue to grow until the threat is described and appropriate safeguards are implemented. Another method of human detection and description of threats is their automatic search. For this purpose, algorithms are used to assess whether the analyzed traffic may pose a threat. The estimation is based on observation and searching for values that do not match the traffic model. The traffic model, which is the reference point for the actual secure operation of the network, can be fixed or time-varying. Observation of anomalies can be based on methods of calculating the probability of the observed event. Periodic or one-off deviations from the standard may also be undesirable, for example, an unusually large volume of traffic exchanged or connections to addresses never before observed in the network. The advantage of these solutions is the ability to detect danger before damage is done. Recognized threats based on traffic observation can be used to create patterns of dangerous events and support signature-based methods. Unfortunately, unlike methods that look for unambiguous patterns, they can generate many false positives and block traffic that is not dangerous. Often, to prevent such situations, allowlists are used, which contain entries identifying traffic sources that are completely safe and, therefore, cannot be recognized as a threat.

Machine learning algorithms are used to detect threats in the network. They allow for processing large datasets and, based on them, create complex classifications of traffic observed in the network.

The aim of the paper is to propose machine learning models using a minimum number of features and to test their effectiveness in detecting Internet threats based on the analysis of network traffic. The work also addresses the issue of detection verification using labeled data containing threatened packets. The results were presented in such a way as to enable comparison of the tested methods. The operation of the algorithms used on various datasets was analyzed. The performance of the models was assessed by comparing the results with other studies, where more features were taken into account. The number of features in all studies is intentionally set to four and remains the same. The features selected by the authors describe each network traffic using the IPv4 protocol. Therefore, it can be widely used in ICT networks, especially in devices at the network's edge, such as Smart Grid. Furthermore, the choice of four features allows for simplifying the traffic analysis models and skipping the preprocessing of Internet traffic records. These two aspects limit the demand for computational resources in devices using the proposed techniques. It is, therefore, possible to use the tested techniques in IoE devices found in the Smart Grid but also in IoT and other devices that do not have extensive computing resources or an external data analysis center.

The work consists of nine sections. Section 2 presents an overview of research carried out so far in the field of threat detection. Section 3 discusses the datasets used, their origin, and properties. The steps taken to prepare the analyzed traffic features properly are the content of Section 4. The preparation of the data for the study and the eight experiments performed are described in Section 5. A comparison of the methods used is included in Section 6. The debate on the results and the detection efficiency of the solutions used are in Section 7. The summary of the work with the final conclusions is presented in Section 8. Section 9 is dedicated to the future of research focused on anomaly detection.

2. State of the Art

Smart Grids could be considered any other ICT network because those are used the same communication protocols. The cyber threat defense based on anomaly detection could be applied to any network based on Internet traffic. Some technics used in IoT could be applicable to IoE and other Smart Grid solutions.

A popular method of analyzing network traffic is Long Short-Term Memory (LSTM). The authors in [7] tested the effectiveness of the LSTM method on the CIC-IDS2017 dataset, which consists of five days of recording the network operation. They conducted traffic class prediction studies for each day separately. Three metrics were used: precision, recall, and F1-score. Very good results were obtained, exceeding 0.98 in each case, which means high efficiency in predicting the traffic class. The method called "Mutual Information" was used to select the analyzed features, indicating the relationship between the two selected parameters [8]. The analysis was based on network flows. The application of the described methods on the selected dataset results in very good traffic classification. Using the LSTM method, researchers from Huawei Technologies and the China University of Geosciences also searched for anomalies in the network. They studied traffic collected from approximately 31,000 ports at five-minute intervals. Then, the task of the model was to classify the data into three defined classes—warning, problem, and alarm [9]. The results were evaluated using the precision and recall metrics. The biggest challenge for the model turned out to be the correct generation of warnings, defined as a single deviation from the norm. Repeated deviations have been called the problem. The alarm was generated by non-standard events that occurred continuously. In their approach, the authors distinguished traffic anomalies, showing that not all should be treated in the same way. The use of more classes is required to distinguish the level of danger of the observed anomalies.

The researchers in [10] showed the impact on the results of matching the categories of various algorithms depending on the number of examined features. Three sets of features

were defined for analysis. The first one was based on the sliding windows technique. The second set was created based on the methods described in [11]. These are Holt Winter methods, Adaptive Threshold Algorithm, Windowed Average, Exponential Moving Average, and Cumulative Sum Algorithm. For the third set, 12 features were selected based on values, statistical metrics, time series, and wavelet decomposition [12]. The Exponentially Weighted Moving-Average (EWMA) method was used to prepare the time series, and autoregression was used [13]. The use of the described methods for feature extraction was possible thanks to the approach based on flow analysis. F1-score was used as a metric. LSTM demonstrated the best results on the third dataset. Compared to other algorithms, it was repeatable. Additionally, Support Vector Machine (SVM), Random Forest (RF), and Adaptive Label Screening and Relearning Approach (ALSR) were checked. The autoregression method used by the authors was also independently used to analyze anomalies in network traffic.

The document [14] shows the use of the Auto-Regresive Moving Average (ARIMA) method to detect network attacks. The method allows calculations to be made on the volume of exchanged traffic. It looks for non-standard values that it considers to be anomalies. The authors present this method as a way to detect Distributed Denial of Service (DDoS) attacks early. They also considered the use of methods presented and described much earlier, including the method called Fractionally Differenced Autoregressive Integrated Moving Average (FARIMA) [15]. It is a moving average-based traffic modeling applicable to the short and long-term prediction of network behavior. Another considered method, also based on the use of a moving average, was the use of Seasonal Autoregressive Integrated Moving Average (SARIMA) modeling described in [16]. The argument for considering these methods by the authors [10] was the periodicity of network traffic. They should then show deviations from the expected behavior of the network. SARIMA and FARIMA modeling look for non-standard values, which it considers as an anomaly. It is well known that the network does not always behave periodically. There are non-standard periods. Then the question remains about using these models in a real network because the detection of threats based on the prediction of network behavior is a method that does not take into account unexpected events.

The GRU (Gated Recurrent Unit) is a method similar to LSTM. Fan et al. tested the use of GRU for network traffic analysis. They also used three metrics different from those previously described: Mean Square Error (MSE), Normalized Mean Square Error (NMSE), and Mean Absolute Relative Error (MARE) [17]. Satisfactory results were obtained, respectively: 0.011, 0.972, and 1.171. However, the cited work lacks a comparison to other analyzes on the same dataset and set of features.

The Support Vector Machine (SVM) is a common method used for research in the area of traffic anomaly. In [18], Yang Lei used only six features for anomaly detection. This allowed for the calculation of entropy, which was the input to the model. Only one metric was used–accuracy. The evaluation of the model was presented based on the effectiveness of detecting various types of threats. The lowest value of the accuracy parameter equal to 0.786 was achieved for a Denial of Service (DoS) attack. The best results did not exceed 0.875. The use of entropy in the study of network traffic can be found in many studies. This allows observing the behavior of the network using a metric for which some standard values can be specified. In 2015, researchers from the Military Institute of Communications and the AGH University of Science and Technology in Krakow studied the use of entropy to detect botnets [19]. Threat detection itself was based on the search for anomalies in traffic. This method is described as a method to look for malware or scams. The authors also found this approach appropriate for fault finding or system monitoring. In this approach, the use of entropy gains an advantage over typical machine learning models. These models, at some point when a failure occurs, could continue to run in the background without showing any signs. Entropy allows monitoring a parameter that exceeds statically or dynamically defined limits and may also indicate faults. In [20], entropy was used to search

for abnormal events to detect DDoS attacks. These studies allow us to conclude that the methods of detecting dangerous events using entropy still need to be refined.

Another way to search for anomalies is through various types of algorithms based on binary and decision trees. The use of tree structures is effective for datasets with a large number of parameters. In [21], the authors were searching for anomalies in the communication of Internet of Things (IoT) devices using a smaller number of parameters. Three sets of features with 15 and 11 parameters, respectively, were used. High efficiency in detecting DDoS attacks has been achieved. The value of the accuracy parameter was as much as 99.94% with the use of the Random Forest algorithm. One should note that in the above-described comparison of algorithms for different sets of features, the results obtained with this method were the best when using a set of 12 features [10]. The authors in [22] studied the Random Forest algorithm in relation to the C4.5 decision tree. Higher efficiency of the decision tree and the accuracy metric value of 99.67 was obtained; however, the RF operation turned out to be much faster. However, decision trees do not always show better results. In a study where decision trees were compared with SVM, worse results were obtained for all sets of features [23]. The measure used was accuracy, and the SVM result was several percentage points higher in all cases. This proves a better fit for the category.

Another researched algorithm is Isolation Forest (IF), i.e., a forest of isolated trees. Studies presented in [24] showed similar IF results to SVM. The number of threats detected by the IF algorithm did not differ significantly from the SVM. For some attacks, the results were worse. However, in most cases, they were slightly better. Research has shown that the classification of both methods results in the correct matching of the analyzed data to the appropriate categories. In [24,25], a comparative study was conducted to extract feature sets using different datasets and different data processing methods. The IF algorithm was used in the study. The results showed greater differences using different datasets within each of the selected methods. Changing the data processing method using a single dataset had a lesser impact on the results. This may indicate that the input to the model based on the same information in the case of this method gives very similar results. An interesting method of preparing features for analysis based on the Kalman filter was used in [26]. The researchers in [27] decided to combine SVM with threat detection systems, analyzing traffic in five steps. The network traffic was processed by Intrusion Detection System (IDS) and then by SVM. This synergy allows for the detection of 70.69% of attacks. The results were compared with those obtained from the SNORT software. The combination of several techniques resulted in more true positives while retaining fewer false positives. The combination of many techniques to detect a threat is an interesting direction to strengthen the network's defense against attacks. The next stage of work in this direction may be treating each other's systems as reliable databases. The machine learning model could learn based on the results of the IDS, and the antivirus software could create signatures based on the results of the analysis made by the machine learning algorithm.

Another commonly used anomaly detection method is Multi-Layer Perception (MLP). In [28], network traffic based on flows was studied. MLP and decision trees on two datasets were used for the analysis. For the dataset named "winter", the detection level was lower using MLP than for the decision tree. The result achieved by MLP was 99.59% of detected threats compared to 99.98% using decision trees. However, when analyzing the second selected dataset, the statistics reversed. The detection ratio obtained with the decision tree was at the level of 88.53%, and for MLP, it was 93.29%, which still this is a very high score. However, the authors in another study proved that MLP generates a worse result than Random Forest [29]. RF turns out to be better than MLP, whose results are better than for decision trees. The difference in the value of the F1-score parameter is around 0.2, which is a significant difference in the classification. In the previously described studies, the results of the RF were worse than the results for the decision tree, which means that it is impossible to say unambiguously which of the methods is the best. Once again, the results showed how important it is to choose the right data. Unfortunately, in a real ICT network, the traffic

is not matched to the model, so the model must be universal enough to work effectively in changing conditions.

Machine learning uses convolutional networks for many applications. Their proper use is the analysis of images and recognizing the elements on them, but they are not used to classify network data. However, in [30], interesting research was published that allows addressing traffic records to Convolutional Neural Network (CNN) models, presenting it in the form of graphics and then searching for threats. The biggest challenge in using this type of neural network is the representation of network traffic as a graphic-like matrix. The encoding described in [31] allows achieving accuracy at the level of 88–89%.

In the field of network analysis, the authors freely select datasets for the needs of their work. It is common to omit the description of the available datasets and focus on working with one selected dataset. The authors in [29] extended the research and presented 11 available datasets that allow data analysis using machine learning techniques. Their work was the basis for the selection of the databases containing traffic records used in this work. The work carried out was based on the described CIC-IDS2017 collection [32]. Other data used in this work come from ASNM resources, where features, composition, and structure of the datasets are documented [33]. These datasets are especially recommended for traffic classification studies.

The analysis of research works consisting of detecting anomalies in Internet traffic shows that the dominant trend is to increase the number of analyzed traffic features or their complicated processing to increase the efficiency of event categorization. In this paper, a new approach for detecting anomalies in ICT networks is proposed. The assumption we had in mind was to minimize the input parameters and simplify their coding. To the best of our knowledge, the analysis we propose is the first in the literature that limits the number of analyzed network traffic features to only four, available in any Internet Protocol (IP) communication, to achieve better performance on devices with limited computing resources as IoE, IoT, or any edge computing devices. We consider two different datasets and the subset of one of them. As presented before, a common approach is to increase the number of analyzed features and use huge computational resources. This paper assumes minimization of the number of features to optimize the resources necessary to classify network traffic. The performed research allows for improving the level of security, data integrity, and confidentiality in smart grid devices on the side of grid operators and the customer. The security of customer data is fundamental, so any solution that could be useful in small smart devices can increase users' trust in the smart grid and, thus, accelerate the implementation of intelligent energy solutions.

3. Datasets

The data used for the analysis are datasets containing information from IP packets. The models work with the following data:

- Source IPv4 address
- Destination IPv4 address
- Source port
- Destination port

Chosen traffic features describe any network traffic that uses the IPv4 protocol. Models based on these four features can be widely used in ICT and Smart Grid networks. At the same time, the choice of the four features indicated reduces the need for pre-processing of the analyzed network traffic. Limiting the number of features also allows for reducing the demand for computational resources.

The selected dataset can be replaced with any dataset containing the above data. An important feature is data labels describing whether the packet contained features indicating the transferred threat. Data categorization is essential to train models properly in a supervised manner and to verify the correctness of all methods used.

3.1. ASNM-CDX-2009

The dataset used comes from the ASNM datasets database. It contains categorized data describing network traffic [34]. It was named ASNM-CDX-2009 and collected by the National Security Agency of the United States of America (NSA) [35]. CDX is named for "Cyber Defense Excercise". The NSA describes the exercise as follows: "The goal of the annual Cyber Defense Exercise (CDX) is to provide a simulated real-world educational exercise that will challenge university students to build secure networks and defend those networks against adversarial attacks" [36]. ASNM, an acronym for Advanced Security Network Metrics, is a set of network data describing TCP connections containing various characteristics. These datasets were created for the needs of traffic analysis, detection, and recognition of threats [37]. The selected dataset contains data on traffic carried out using the TCP protocol. The original CDX-2009 collection contains approximately four million entries but has been limited by ASNM originators to 5771 categorized connections as presented in Table 1.

Table 1. Network traffic statistics from ASNM-CDX-2009.

Network Service	Number of TCP Connections		
	Safe	Unsafe	Total
Apache	2911	37	2948
Postfix	179	7	186
Other traffic	2637	n/a	2637
Summary	5727	44	5771

Entries in the ASNM-CDX-2009 file are categorized and do not contain data carried in IP packets. They are described by two labels:

- "label_2" indicates whether the entry concerns a buffer overflow attack
- "label_poly" has a binary-descriptive structure. The first part informs whether the traffic is safe or unsafe, with the values zero and one, respectively. The second part indicates the service related to the traffic. Three service descriptions are defined: apache, postfix, and other. An example entry is 0_postfix, indicating that the entry is for a secure connection and email service.

Data from the ASNM-CDX-2009 database are available for download from the Internet [38]. The records of network traffic, on the basis of which the CDX-2009 dataset was created, are also publicly available [39].

3.2. CIC-IDS2017

The dataset contains observations for five days—Monday through Friday. The traffic is generated and analyzed in the test topology described by the authors in [29]. It is limited to the proposed topology and is well-described. There are versions with traffic records in the form of PCAP files and extracted with many features in the form of Comma-Separated Values (CSV) files [32]. A more convenient form for analysis is a CSV file, so data in this form was used in this work as the input to the models. The data includes seven types of attacks:

- Brute force attack—discovers passwords or hidden resources on the basis of making many attempts,
- Heartbleed attack—based on the search for imperfections in the encryption of network traffic,
- Botnet—a network of infected devices used by cybercriminals,
- Denial of Service (DoS) attack—an attack that overloads the infrastructure to prevent it from working properly,

- Distributed Denial of Service (DDoS) attack—a method similar to DoS, however, the attack is carried out by many distributed devices at the same time, the goal remains the same as the goal of the DoS attack,
- Web attack against WWW services—based on searching for their weak points,
- Infiltration attack—exploiting victims' software vulnerabilities to search their internal networks.

The entire dataset consisting of eight CSV files contains 3,119,345 entries, of which 846,248 were marked as dangerous (which is 27.13% of all entries) and 2,273,097 as not posing a threat (which is 72.87% of all entries). The types of attacks in the dataset are properly marked, which allows for extensive analysis to classify attacks into specific categories. For the purposes of the research conducted in this paper, information about attacks will be binary-encoded, as the goal is to detect the threat quickly.

4. Data Preparation

Proper data preparation is a particular challenge. Encoding IPv4 addresses turn out to be a non-trivial task. This is due to a large number of available addresses, for which encoding becomes a computationally difficult problem. Encoding all possible IPv4 addresses involves building a vast dataset that needs to be queried every time we want to categorize traffic. The operation must be performed twice. Once for the source address and once for the destination address. In the case of also analyzing addresses of physical network interfaces, the problem becomes even more complex.

An IPv4 address consists of four octets. Apart from the division into subnets, it can be said that there are $2^{32} - 2$ of all available addresses, although packets sent to a broadcast address can also be found in the network. This means that $2^{32} - 1$ different combinations must be encoded. The size of the array mapping addresses to labels would be huge (1).

$$2^{32} - 1 = 4294967295 \quad (1)$$

Using one-hot coding will be highly inefficient because the dataset in the model would be extended by two square matrices with dimensions corresponding to the number of addresses. Frequency coding is not possible because each address occurs only once. Each entry will, therefore, be presented as exactly the same number. It becomes obvious that the set of addresses used should be limited. A certain form of limitation is the exclusion of addresses that should not appear in a given network segment. For the analysis carried out in the public network, these will be addresses from private pools:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

A restriction of this type will reduce the number of entries needed to be encoded. The pool of excluded addresses is, unfortunately, so small that it does not solve the problem. Another solution may be to restrict operation to a certain private network. This approach obscures information about the real source address, thus limiting the possibility of detecting threats. Due to the above-described problems, this work focuses on the ready dataset and encodes only the addresses that appear there.

Physical addresses are also included in the analyzed data. Their coding causes an even greater computational problem. Due to two more octets. There are 2^{48} total MAC addresses. This aspect is facilitated by the fact that the number of physical addresses in the node where the traffic is recorded is very limited. These are only the nearest, directly connected neighbors. Depending on the point in the network where the traffic is registered, these may also be all interfaces present in a given subnet or within the range of a wireless device. The drawback to this simplification is the problem of unique traffic characteristics since MAC addresses will always point to directly connected devices. When listening between two routers, the MAC addresses will always remain the same. The problem with encoding grows to a whole new dimension when using IPv6, where addresses are as large

as 128 bits. This is four times more than in the case of IPv4, which means that it significantly increases the number of encoded entries and requires even more resources for analysis. The set described in the previous chapter does not contain IPv6 addresses, so this problem is omitted in this paper.

5. Results

Data analysis is divided into three subsections based on datasets. The first describes an application of procedures to the ASNM-CDX-2009 dataset, the second to the CIC-IDS2017 dataset, and the third to a single day from the CIC-IDS2017 dataset. Sections 5.2 and 5.3 describe why the CIC-IDS2017 dataset is used in two steps. As mentioned before, four features were analyzed from all chosen datasets (Table 2).

Table 2. Features from datasets.

Feature	Coulmn in ASNM-CDX-2009	Column in CIC-IDS2017
Source IPv4 address	SrcIP	Source IP
Destination IPv4 address	DstIP	Destination IP
Source port	SrcPort	Source Port
Destination port	DstPort	Destination Port

Metrics used to evaluate models are:

- Precision–Fraction of true positive to the sum of true positive and true negative prediction (2);

$$Precision = \frac{TP}{TP + TN} \tag{2}$$

- Recall–Fraction of true positive to the sum of true positive and false negative prediction (3);

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- F1-score–Fraction of the product of precision and recall to the sum of precision and recall (4).

$$F1 - score = \frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{4}$$

The summary score will be calculated by averaging the F1-score parameters. There is a parameter called "macro average F1-score".

5.1. Analysis of ASNM-CDX-2009 Dataset

ASNM-CDX-2009 is a labeled dataset, but the packet's label and service, related to data, get through into one column called "label_poly". So, the dataset needs extra work before analyzing it in models. Column "label_poly" was divided into two columns—"label" and "poly". The information about the services correlated with packets contained in the "poly" column is unnecessary. The "label" column contains values 0 or 1, where 0 is secure, and 1 is a threat.

The encoding method used is label encoding. Only the IPv4 addresses present in the dataset have been replaced. The port numbers are integers, therefore, did not require any coding. In the procedure related to the division of the "label_poly" column, labels occurred in binary form and, therefore, did not require coding.

5.1.1. Long Short-Term Memory Classification

The model was designed using the LSTM method. The training data constitute half of the ASNM-CDX-2009 dataset. Three layers of LSTM were used, separated by layers responsible for regularization to prevent overtraining; these are dropout layers [40,41]. At

the model's last layer, a Dense type layer was used to ensure an appropriate output [42]. Thus, the model was built of exactly eight elements (Figure 1).

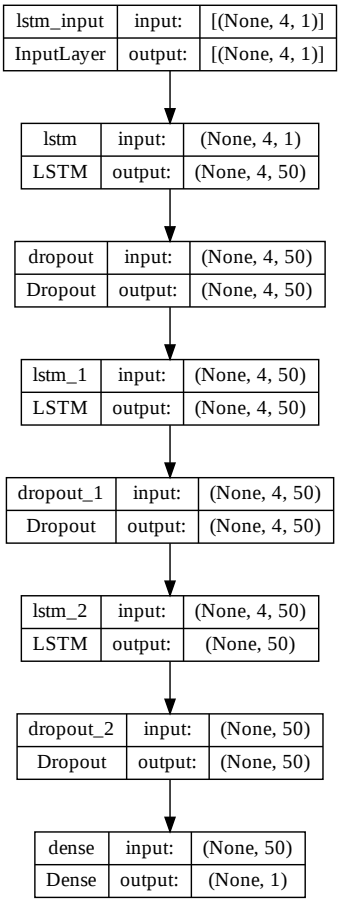

Figure 1. The multi-layer construction of the Long Short-Term Memory model.

The numbers next to the layers indicate the dimensionality of the data. The commonly used optimization method "Adam" (Adaptive Moment Estimation) was used [43]. "mean_squared_error" was used as a metric determining the effectiveness of the model [44]. Other metrics considered are "mean_absolute_error" and "binary_crossentropy" [44,45].

The mean square error is the average of the squares of the difference between the expected value and the result of a given trial (5).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_{Pi} - Y_{Ti})^2 \quad (5)$$

The mean square error is the average of the absolute difference between the expected value and the result of a given trial (6).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_{Pi} - Y_{Ti}| \quad (6)$$

The binary cross entropy error is the average of the sum of the product of the expected value and the logarithm of the result of a given trial and the product of the difference between one and the expected value and the logarithm of the difference between one and the result of a given trial (7).

$$H = -\frac{1}{N}\sum_{i=1}^{N} Y_{Ti} \cdot log(Y_{Pi}) + (1 - Y_{Ti}) \cdot log(1 - Y_{Pi}) \quad (7)$$

The number of epochs was set at 30, which means the number of passes of the training data through the entire model. A correctly compiled model is one where the "loss" value for the metric used shows a decreasing direction with successive epochs. Training the model using the binary cross entropy loss function shows a sudden drop followed by fluctuations. The training process was observed correctly for the loss functions' mean absolute error and mean squared error because the values fall according to successive epochs (Figure 2). In further consideration, only the results of the model trained using the mean squared error function are taken into account because the highest efficiency is characterized in this model.

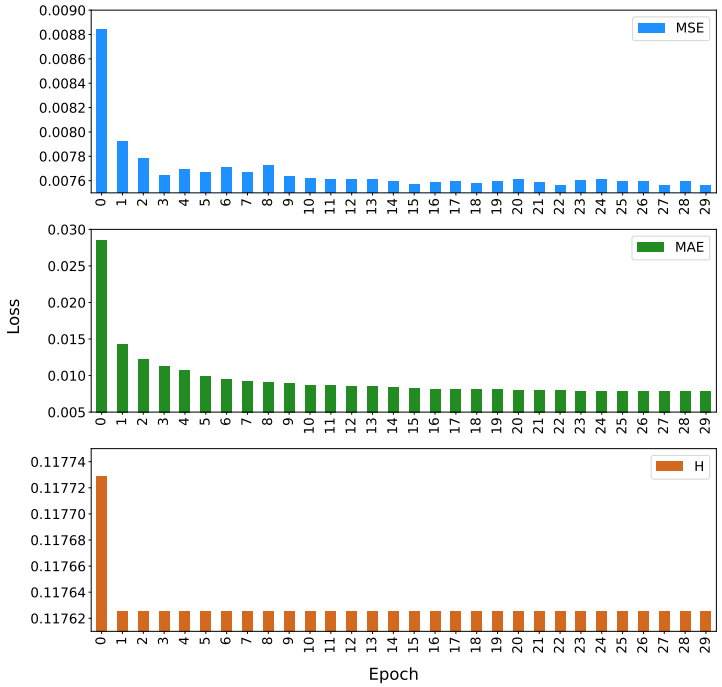

Figure 2. Long Short-Term Memory loss parameter by epochs based on the ASNM-CDX-2009 dataset.

Verification of the model with validation data, which constitute the second half of the ASNM-CDX-2009 dataset, allowed collection of the results of its effectiveness.

5.1.2. Isolation Forest Classification

Isolation Forest (IF) is an algorithm for anomaly detection based on binary trees. The algorithm could realize analysis without having labeled data. So, it is possible to use any dataset. However, labels are obligatory to validate the analysis. For the proposed research, the number of single isolation trees was set to one hundred.

The analysis using the IF algorithm showed that the number of packets marked as anomaly was extremely high–5482. On the other hand, in the dataset, only 44 packets were recognized as suspicious. The number of secure packets was only 289 after IF analysis, but in the ASNM-CDX-2009 dataset, 5727 records were labeled as secure. So, the IF algorithm made the wrong categorization.

After receiving the results shown above, a reverse analysis of the IF algorithm was also used. The implementation of reverse categorization allowed a more similar evaluation of the dataset to assigned labels. Reversed analysis computed that 5482 packets are secure, and 289 packets are risky. Thus, the results are exactly the mirror image of those previously reported.

5.1.3. Support Vector Machine Classification

In this part of the work, an analysis model based on the SVM was developed. Four different versions of the kernel function were used: linear, polynomial, sigmoid, and Radial Basis Function [46]. The division of the dataset was applied the same as for the LSTM model. The research procedure gave the same outcome for every used kernel function.

5.1.4. Summary of the ASNM-CDX-2009 Dataset Analysis

The effectiveness of the described methods is very weak for the ASNM-CDX-2009 dataset. It should be noted that the number of features has been limited to only four. Good recognition of safety packets provides SVM and Reversed Isolation Forest (RIF), but the correctness of diagnosis threats is close to 0% (Table 3). Other methods such as LSTM and IF gave inaccurate results too.

Table 3. The models' metrics after validation for a single day from the ASNM-CDX-2009 dataset.

Method	Packet Type	Precision	Recall	F1-Score	Support
LSTM	0	1.00	0.08	0.14	2863
	1	0.01	1.00	0.02	22
SVM	0	0.99	1.00	1.00	2863
	1	0.00	0.00	0.00	22
Isolation Forest	0	1.00	0.05	0.10	5727
	1	0.01	1.00	0.02	44
Reverse Isolation Forest	0	0.99	0.95	0.97	5727
	1	0.00	0.00	0.00	44

5.2. Analysis of CIC-IDS2017 Dataset

Dataset CIC-IDS2017 is divided into eight files, but "Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv" was omitted because of the encoding issues. As a result, the analyzed part of the CIC-IDS2017 dataset had 2,660,377 records—555,466 marked as dangerous and 2,104,911 labeled as secure packets.

IPv4 addresses were encoded using Label Encoding. The statuses of the data packets were binary-coded on the principle that a packet containing a threat is 1, and a safe one is 0. Due to a large amount of data to be analyzed, the SVM method was highly inefficient, making it impossible to carry out the analysis in a limited time using the available resources.

5.2.1. Long Short-Term Memory Classification

Data before analysis were randomly mixed in an automatic manner [47]. Then the dataset was divided into two parts: training and validation data. The parameters of the model, as well as its construction, remained unchanged compared to the previous analyzes described in Section 5.1.1. However, higher values of the "loss" parameter were observed for every used function. The model training process was correct only for the mean squared error loss function (Figure 3). Using the binary cross entropy and mead absolute error metrics, the value of the "loss" parameter increases significantly in the final stage of training, which is an undesirable phenomenon. Therefore, as in the case of the analysis of the ASNM-CDX-2009 data set, only the results of the model based on training with the use of the mean squared error loss function were qualified for further consideration.

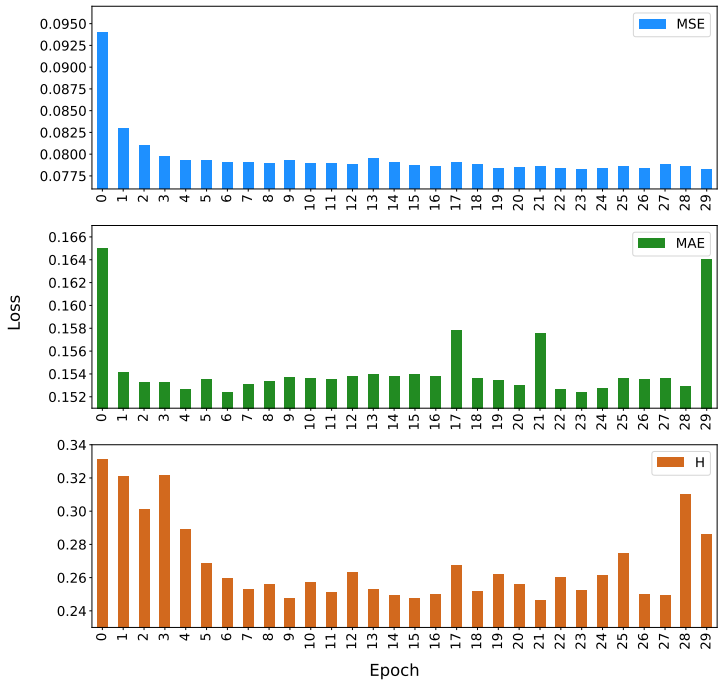

Figure 3. Long Short-Term Memory loss parameter by epochs based on the CIC-IDS2017 dataset.

5.2.2. Isolation Forest Classification

The configuration of the model from Section 5.1.2 was repeated. The analysis was performed again, interpreting the results in two ways, standard and reversed.

The number of packets classified as anomalous events was 2,527,358. That vast number is evidence that the model's interpretation was wrong because the number of labeled packets as dangerous in the dataset is near to five times less. Packets marked as safe were in the minority—133,019 entries.

The result of Reversed Isolation Forest analysis is closer to dataset statistics. In the dataset, 555,466 are marked as dangerous, but IF analysis presents 133,019 risky records and 2,527,358 safe packets.

5.2.3. Summary of CIC-IDS2017 Dataset Analysis

The described methods' effectiveness based on the metrics collected for the CIC-IDS2017 dataset can be considered much better than the ASNM-CDX-2009 dataset. The size of the dataset affects the effectiveness of the selected methods. However, the results are still not satisfactory. RIF analysis showed the best results, but only when identifying secure packets (Table 4).

Table 4. The models' metrics after validation for the CIC-IDS2017 dataset.

Method	Packet Type	Precision	Recall	F1-Score	Support
LSTM	0	0.79	0.33	0.47	1,052,095
	1	0.21	0.67	0.32	278,093
Isolation Forest	0	0.79	0.05	0.09	2,104,911
	1	0.21	0.95	0.34	555,466
Reverse Isolation Forest	0	0.79	0.95	0.86	2,104,911
	1	0.21	0.05	0.08	555,466

5.3. Analysis of Selected Part of the CIC-IDS2017 Dataset

The volume of data was reduced to one day due to the size of the CIC-IDS2017 dataset and according to the availability of the resources. Friday was indicated as input to models because of the best balance of secure and risky records. The number of secure packets is 414,322 (59% of packets), and 288,923 (41% of packets) are labeled dangerous. The sum of events in files associated with Friday is 703,245. For example, in the validated CIC-IDS2017 dataset, 79% of packets are marked as secure and 21% as dangerous.

5.3.1. Long Short-Term Memory Classification

Data preparation has not changed concerning the operations described in Section 5.2.1. The construction of the model and all its parameters were left unchanged compared to those described in Section 5.1.1. It was observed that the "loss" parameter drops sharply and stabilizes for a long time during training the model by the mean squared error function (Figure 4). Despite fluctuations, a version of the model using the loss binary cross-entropy function can be considered adequately trained. The results for the model using the mean absolute error metric showed significantly lower efficiency than the other models. Comparing the model trained with binary cross entropy and mean squared error, the macro average F1-score was better for the model using binary cross entropy—0.48. For the model using mean squared error, it was 0.34. However, the F1-score for the packets at risk category was higher for the model trained with mean squared error—0.57 than for the model trained with binary cross-entropy; it was 0.49. Thus, as in Sections 5.1.1 and 5.2.1, it was decided to conduct further analysis of the model version trained by the mean squared error loss function.

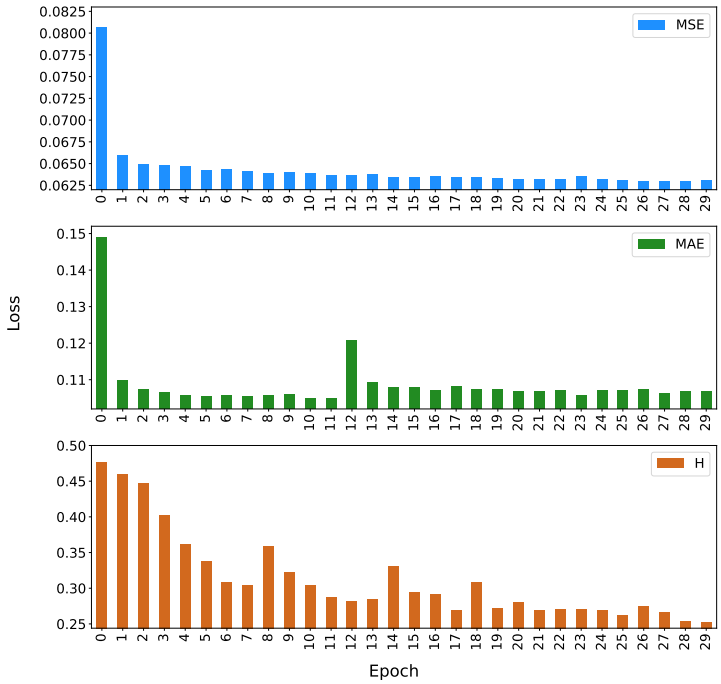

Figure 4. Long Short-Term Memory loss parameter by epochs based on a single day from the CIC-IDS2017 dataset.

5.3.2. Isolation Forest Classification

The steps described in Section 5.2.2 were repeated. The collected results have a similar division into categories to those previously observed, using IF for classification. The huge number, 668,082 packets marked as risky, and only 35,163 were marked as secure. The result is entirely inconsistent with the labels in the dataset.

After reversing the analysis, we observed a deterioration in detecting dangerous packets—35,163 packets were marked as an anomaly and 668,082 as typical network traffic.

The distribution of packets classified into two categories is far from the correct distribution of the analyzed dataset.

5.3.3. Support Vector Machine Classification

After limiting the dataset to one day, the analysis results were collected using an SVM. The observed results are much better than those using the ASNM-CDX-2009 dataset. Increasing the dataset has a positive effect on the results of the described method. The model options relative to operation in Section 5.1.3 have not been changed. The analysis was the most time-consuming of all the described methods. The Radial Basis Function was used. A significant improvement in statistics was observed compared to operations performed on the previously analyzed smaller dataset.

5.3.4. Summary–Selected Part of the CIC-IDS2017 Dataset

Analysis using Isolation Forest is useless, but all other methods gave interesting results (Table 5). LSTM and SVM work better when the training dataset is balanced. Adequate and balanced training data can significantly impact the learning process and, therefore, the accuracy of detecting potentially dangerous events. Moreover, the dataset used to train the model should be large enough to carry out the training process properly. However, the dataset must be constrained before analysis for model training and their application to be feasible.

Table 5. The models' metrics after validation for a single day from the CIC-IDS2017 dataset.

Method	Packet Type	Precision	Recall	F1-Score	Support
LSTM	0	0.59	0.06	0.11	206,955
	1	0.41	0.94	0.57	144,667
SVM	0	0.59	0.69	0.64	207,147
	1	0.41	0.31	0.35	144,475
Isolation Forest	0	0.53	0.05	0.08	414,322
	1	0.41	0.94	0.57	288,923
Reverse Isolation Forest	0	0.59	0.95	0.73	414,322
	1	0.47	0.06	0.10	288,923

6. Results Comparison

For each method and dataset, a macro average F1-score was calculated to compare models for different data.

6.1. Long Short-Term Memory Comparison

The Long Short-Term Memory method achieved the highest effectiveness in detecting threats for a large unbalanced set of data (Table 6). More data may make LSTM more effective, and the balance between safe and dangerous packets above some size of the dataset is less important. The analysis of selected data sets using the LSTM method shows a significant impact of the selection of the loss function on the classification results. In the examined problem, it could be assumed that models trained using functions for binary classification, such as binary cross-entropy, will show higher effectiveness. However, an important observation is that models trained using the mean squared error loss function demonstrated the highest effectiveness of unsafe event identification.

Table 6. Long Short-Term Memory results comparison for all datasets.

Dataset	F1-Score
ASNM-CDX-2009	0.08
CICIDS-2017	0.42
Friday–CICIDS-2017	0.34

6.2. Isolation Forest Comparison

The highest efficiency in detecting traffic was demonstrated by the inverted analysis on the smallest dataset, a small part of which are at-risk packets (Table 7).

Table 7. Isolation Forest algorithm results comparison for all datasets.

Dataset	F1-Score	
	Basic	Reverse
ASNM-CDX-2009	0.06	0.49
CICIDS-2017	0.22	0.47
Friday–CICIDS-2017	0.33	0.42

These results indicate that the more balanced the data, the better the performance of the IF algorithm. Furthermore, the efficiency of network traffic recognition increases when an anomaly is considered a safe event and packets marked as safe are considered unsafe. The reversed interpretation of the results is used, which is a better option for anomaly detection using the ASNM-CDX-2009 dataset, the CIC-IDS2017 dataset, and its subset.

6.3. Support Vector Machine Comparison

Comparing the results, the Support Vector Machine has the greatest efficiency in identification. The best result was obtained for the smallest dataset, where the detection of threats was close to or equal to zero. The main component of this result is the recognition of safe network traffic, which is not the essence of the research. Despite the lower value, the data analysis from a larger dataset turned out to be better (Table 8). The better result could also be influenced by closer to an equal balance of the input to the model.

Table 8. Support Vector Machine results comparison for all datasets.

Dataset	F1-Score
ASNM-CDX-2009	0.50
Friday–CICIDS-2017	0.49

7. Discussion

The results presented in the previous chapter are a summary of the analysis of data from publicly available datasets. The use of marked records was necessary for the correct verification of the operation of the models. Although SVM's performance is the best in the analyzed cases, the best results were achieved for LSTM. This is due to better recognition of threats, which is much more important than classifying safe packets. The LSTM algorithm is definitely faster in operation. The learning process itself takes time, but verification of validation data was the fastest of the methods tested. Long Short-Term Memory also shows very good results in other studies. The results with a very high F1-score collected by the authors of the analysis conducted on the CIC-IDS2017 dataset in [7] show that the high efficiency of this model is achievable on selected data (see Table 9).

Table 9. Other results based on the CIC-IDS2017 dataset [7].

Day of the Week	F1-Score
Tuesday	0.989
Wednesday	0.992
Thursday	0.985
Friday	0.991

It can be seen that the result achieved for Friday's data is much better, 0.991, compared to 0.39 achieved in the conducted research. It should be noted that the authors of [7] studied flows, not individual packets, as in this paper. The study of flows in the network simplifies the search for anomalies due to the very large resources of data describing the flow. Each flow includes at least a few packets that describe a networking event, and each packet has many characteristics of its own. The study used a much larger number of traffic features than four. This causes another increase in the amount of data describing the traffic. A very large amount of information describing the event allows us to identify it better and, thus, also creates more precise recognition structures.

The research conducted as part of this work was guided by the minimization of the analyzed features. The four features of network traffic used in the experiments described in two previous chapters greatly simplify data processing and make the model easily adaptable to other datasets. This approach facilitates its wide application and the possibility of examining traffic on less efficient devices at the edge of the network (e.g., IoT or IoE devices). The number of analyzed features has a significant impact on the results of the models. The study conducted by the authors in [11] on different sets of features of selected data shows differences in effectiveness depending on the number of features. How much a set of features changes in network traffic analysis can be seen in the comparative results of four different models for three sets of features (see Table 10).

Table 10. F1-scores for different feature sets [11].

	Feature Set 1	Feature Set 2	Feature Set 3
SVM	0.068554	0.566726	0.618743
RF	0.847268	0.760565	0.877893
LSTM	0.644807	0.659509	0.894281
ALSR	0.808177	0.654177	0.965109

The construction of the first set was based on the sliding windows technique. The second set was constructed using the following methods: Holt Winter, adaptive threshold algorithm, average over time windows, exponential moving average, and cumulative sum algorithm. The third set consists of 12 features that have been prepared on the basis of values, statistical metrics, time series, and wavelet decomposition. Despite the use of advanced methods of collecting data features, one of the SVM results turned out to be much worse than in the methodology adopted in this work, assuming the simplification of the model to four features. Analysis using LSTM on the CIC-IDS2017 dataset shows differences in the F1-score from about 0.21 to 0.46. The maximum difference shows the definite differences in the classification efficiency of the two approaches. However, the smallest differences show that the model developed during the experiments conducted in this work is characterized by good performance when minimizing the input data. The method based on a strong simplification implies easy adaptation of the model to various data. The GRU (Gated Recurrent Unit) method, similar to LSTM, was used to measure the mean square error [48]. A result of 0.011 was achieved [17]. This is almost twice as high as the results obtained with the ANSM-CDX-2009 dataset but lower than the other studies (approx. 0.062 vs. one-day data and 0.064 vs. the entire available CIC-IDS2017 dataset, respectively). The smallest of the analyzed sets had better error results, probably due to the small number of infected packets, which significantly reduces the possibility of false hits

when a packet is considered safe. The collected results of the operation of the models on various sets present their effectiveness in recognizing traffic (see Table 11).

Table 11. F1-scores of all tested models.

	ASNM-CDX-2009	CIC-IDS2017	Friday–CIC-IDS2017
LSTM	0.08	0.42	0.34
Isolation Forest	0.06	0.22	0.33
SVM	0.50	—	0.49

Despite the seemingly highest efficiency of SVM on the ASNM-CDX-2009 dataset, it can be assumed that this result is unreliable due to the large differences in the number of packets marked as safe and unsafe. A very low number of infected packets in the validation dataset will result in good results even when all traffic is considered safe. This theory is confirmed by the results from Section 5.1.3, where the metrics for dangerous packets–labeled "1", are equal to zero. This means no threats identification. The results of the other methods indicate that when the data is unbalanced, it is difficult to identify vulnerable packets. For reverse categorization using an IF, the detection of suspicious network traffic dropped to a level equal to zero. While maintaining the standard classification, the result consisted of more false classifications than in the case of LSTM. Thus, the first algorithm in the table for relatively small, unbalanced data with a limited number of features will prove to be the best solution. For large, varied datasets, Long Short-Term Memory again proves to be more effective, achieving almost twice as good a result as IF. The use of reverse classification results in very good recognition by the IF safe packets constituting the majority of the examined set, which is presented in the results contained in Section 5.2.2. However, the result of the standard classification should be considered better. Generating a lot of false positives from a security point of view can be considered a better scenario than limited threat detection. The last analyzed set was a subset of the CIC-IDS2017 dataset. This collection was characterized by the best balance. The best result was obtained using the SVM, which was the most effective for the classification of the analyzed traffic. However, as the detailed results in Section 5.3 show, unsafe packets were better detected using LSTM. This means that despite the overall higher SVM performance, fewer threats were detected. Similar results to LSTM were achieved using the IF algorithm. This may mean that the key to the correct operation of this tree structure is to spread the data evenly among the categories. Despite the better results achieved, the results presented in Section 5.3.2, show the distribution of results far from the actual division into categories of the dataset.

8. Conclusions

The work was related to the analysis of threats detection ICT networks, including smart grids based on network traffic analysis. Various datasets with significantly different frequencies of occurrence of threats were examined. The obtained results, despite the fact that they present lower values than in other studies, show the possibility of classifying traffic with a minimum of information about its source and purpose. Unfortunately, traffic classification alone will not help to ensure greater security for network users. In this case, it is more important to identify threats, even at the cost of errors, quickly, and evaluate safe packets as unsafe. The selected day from the CIC-IDS2017 dataset turned out to be the best for the analysis, which may indicate that its balancing has a positive impact on the categorization of traffic. Thus, by expanding the set of training data for model preparation, better results will not always be achieved. At the same time, it can be seen that data containing a limited number of entries will not effectively identify the flow of traffic. For the purposes of applying the models in practice, large amounts of data marked by other systems are necessary to look for characteristic features in future traffic and identify threats in time. The best of the analyzed is the Long Short-Term Memory algorithm, which, despite the requirement of supervised learning, allows one to achieve optimal results. This method

works best when working with different sets of information. The duration of its operation turned out to be the shortest of all tested methods. For data where the number of packets divided into categories is close to each other, the IF algorithm turned out to be a good method. The advantage of IF over LSTM is that it does not require labeled data to function properly. This means that the model can work on data collected directly from the network without using any other scanning method.

Network traffic analysis probably will not protect users from social engineering attacks, but it can help protect companies from leaking their data or from using their infrastructure for purposes inconsistent with their intended purpose, and often even against illegal practices. For home or smart grid users, traffic analysis techniques will help protect their personal devices from malware, adware, and other attacks that use external servers. Network traffic analysis can also help detect the use of home or business IoE/IoT devices to create dangerous botnets. The unusual behavior of customers using corporate network resources is also a potential threat that requires identification using methods based on defined rules and signatures. The potential use of devices in Smart Grid networks for attacks on standard ICT systems is also a threat to Smart Grids because, as a source of dangerous traffic, they can be blocked by other critical infrastructure systems. The possible excessive use of their computing power by cybercriminals increases their operation costs. It also causes a delay in performing the tasks for which these devices are designed, which can harm the operation of the grid. For example, frequent changes of statuses transferred to the Smart Grid may cause problems in the operation of the infrastructure; some of them may be deliberate actions to the detriment of the network, therefore, should be classified as an attack. An important aspect is that all methods used should focus on detecting dangerous traffic, not always anomalies. The challenge is to classify such anomalies properly and not always consider them as threats, as this could lead to problems related to the effective use of the network.

Another important aspect may be the problem of resources. The computational requirements of machine learning methods are very high. Implementation of such methods by home users can be a big challenge. That is why it is so important to simplify models to save resources and energy. Protecting a user from a serious attack may be worth the cost. In most cases, however, there are harmless infections or unauthorized use of user devices. This does not always cause noticeable problems for victims, and then the increase in security maintenance costs may seem unjustified to many people. Cost-effective solutions are also ecological, which may convince more people to use modern security methods. Optimizing energy consumption in devices controlling the energy infrastructure is even more critical because it reduces the cost of maintaining the Smart Grid. All this justifies the purpose of this work to achieve the best results in detecting threats while reducing the complexity of the research conducted. Reducing the level of complexity of the conducted experiments is a challenge in itself. However, as shown in the paper, the search for simple and effective methods to optimally use the available data and resources causes many complications.

9. Future Directions

The development of the described methods can go in many directions. One can achieve higher effectiveness, and as other studies show, it is possible. However, this requires a more extensive analysis of the input data and an increasing number of parameters. Such activities require adequate resources in the form of computing power, energy, and time. On the other hand, the direction of development of the proposed solutions may be research aimed at increasing the effectiveness of detecting threats based on the presented assumptions of limiting the number of features, but by proposing other models or redesigning the structure of the proposed models. Another way may be the study of other features that can be easily adapted and verify whether such an approach will improve the effectiveness of identifying threats. Undertaking further experiments to refine the models can bring measurable results and improve detection, but as described in previous chapters, the data analysis is very important: traffic records, their size, but also the number and type of features selected. Address encoding

methods remain an important aspect, as it is a complex problem that requires resources and appropriately developed solutions to be able to use them on a large scale. Research on the presentation of addresses for models may constitute a large part of future work. The next stage of research is the development of appropriate sets of features that will be universal to such an extent as to maintain the simplicity of implementation while allowing for better threat detection results. The processing of network traffic directly coming from various sources and the preparation of new datasets are also possible. In addition, to better assess the effectiveness of the proposed methods, more data sets registered in various network nodes can be used. The developed methods would allow recording traffic in any network and then conducting its analysis. The consequence of this is the implementation of the described methods to work in real time. The real-time analysis will allow users to be warned in time. This is especially crucial for smart grid users. To increase the efficiency of the learning process, a possible development milestone may be the use of information from multiple nodes using shared traffic records. The mechanism of sharing information about traffic would allow the identification of threats wherever there is no knowledge of their existence. Subsequent nodes can provide each other with new data for training subsequent models or training existing ones. Correct detection of events with features inconsistent with those identified as safe in one node may be the basis of its training set in another.

The use of machine learning methods in the field of network security is undeniable. However, an ongoing problem is the representation of the data. In this work, coding using labels were used. In a real network, an efficient network address coding system would have to be developed. The standard "Label Encoder" encodes the data prior to analysis, which requires to have an input dataset before the model can run and make traffic predictions. This way of representing data requires a lot of resources. In the case when the analyzer works on a real Internet network, there is a possibility of any address. This makes it necessary to encode all addresses in the network and store this data in the device's memory. Developing a universal address translation method would allow the model to work efficiently and limit analyzed features. The use of complex input data representation methods is a common technique. However, machine learning algorithms do not try to understand the meaning of the transmitted data, they search for patterns, and the data representation itself, as long as it is numerical, remains secondary to the algorithm. Such an approach allows searching for new parameters on the basis of which data can be analyzed and categorized. Even using basic traffic information, new features can be developed. An example would be the absolute value between numerically represented destination and source port numbers [49]. Each such parameter describes the traffic and brings a new value to the model. The application of mathematical operations on the basic set of features will allow obtaining a new, more elaborate, or written differently description of the event. By modifying the data representation, the distance between events is affected. From the model's perspective, this can have both negative and positive effects on the classification results.

Author Contributions: Conceptualization, S.S. and M.N.; methodology, S.S. and M.N.; software, S.S.; validation, S.S.; formal analysis, S.S. and M.N.; investigation, S.S. and M.N.; writing—original draft preparation, S.S. and M.N.; writing—review and editing, S.S. and M.N.; visualization, S.S.; supervision, M.N.; project administration, M.N.; funding acquisition, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Polish Ministry of Science and Higher Education with the subvention funds of the Faculty of Computer Science, Electronics and Telecommunications of AGH University of Science and Technology and partially supported by the National Centre for Research and Development, grant number CYBERSECIDENT/381319/II/NCBR/2018 on "The federal cyberspace threat detection and response system" (acronym DET-RES) as part of the second competition of the CyberSecIdent Research and Development Program–Cybersecurity and e-Identity.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. SonicWall Inc. 2022 SonicWall Cyber Threat Report. Available online: https://www.sonicwall.com/resources/white-papers/2022-sonicwall-cyber-threat-report/ (accessed on 5 December 2022).
2. SonicWall Inc. 2020 SonicWall Cyber Threat Report. Available online: https://www.sonicwall.com/resources/white-papers/2020-sonicwall-cyber-threat-report/ (accessed on 5 December 2022).
3. SonicWall Inc. 2021 SonicWall Cyber Threat Report. Available online: https://www.sonicwall.com/resources/white-papers/2021-sonicwall-cyber-threat-report/ (accessed on 5 December 2022).
4. Ding, J.; Qammar, A.; Zhang, Z.; Karim, A.; Ning, H. Cyber Threats to Smart Grids: Review, Taxonomy, Potential Solutions, and Future Directions. *Energies* **2022**, *15*, 6799. [CrossRef]
5. Industroyer2 Malware Targeting Ukrainian Energy Company. Available online: https://www.ironnet.com/blog/industroyer2-malware-targeting-ukrainian-energy-company (accessed on 5 December 2022).
6. Kafle, Y.R.; Mahmud, K.; Morsalin, S.; Town, G.E. Towards an internet of energy. In Proceedings of the 2016 IEEE International Conference on Power System Technology (POWERCON), Wollongong, NSW, Australia, 28 September–1 October 2016; pp. 1–6. [CrossRef]
7. Shi, Z.; Li, J.; Wu, C.; Li, J. DeepWindow: An Efficient Method for Online Network Traffic Anomaly Detection. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 10–12 August 2019; pp. 2403–2408. [CrossRef]
8. Wang, L.; Wang, Y.; Chang, Q. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods* **2016**, *111*, 21–31. [CrossRef] [PubMed]
9. Qin, G.; Chen, Y.; Lin, Y.X. Anomaly Detection Using LSTM in IP Networks. In Proceedings of the 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD), Lanzhou, China, 12–15 August 2018; pp. 334–337. [CrossRef]
10. Jing, Y.; Qi, Q.; Wang, J.; Feng, T.; Liao, J. ALSR: An Adaptive Label Screening and Relearning Approach for Anomaly Detection. In Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; pp. 1–6. [CrossRef]
11. Shanbhag, S.; Wolf, T. Accurate anomaly detection through parallelism. *IEEE Netw.* **2009**, *23*, 22–28. . MNET.2009.4804320. [CrossRef]
12. Lu, W.; Ghorbani, A.A. Network Anomaly Detection Based on Wavelet Analysis. *EURASIP J. Adv. Signal Process* **2009**, *2009*, 837601. [CrossRef]
13. Krishnamurthy, B.; Sen, S.; Zhang, Y.; Chen, Y. Sketch-Based Change Detection: Methods, Evaluation, and Applications. In Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, Miami Beach, FL, USA, 27–29 October 2003; IMC '03; Association for Computing Machinery: New York, NY, USA, 2003; pp. 234–247. [CrossRef]
14. Yaacob, A.H.; Tan, I.K.; Chien, S.F.; Tan, H.K. ARIMA Based Network Anomaly Detection. In Proceedings of the 2010 Second International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 205–209. [CrossRef]
15. Shu, Y.; Jin, Z.; Zhang, L.; Wang, L.; Yang, O. Traffic prediction using FARIMA models. In Proceedings of the 1999 IEEE International Conference on Communications (Cat. No. 99CH36311), Vancouver, BC, Canada, 6–10 June 1999; Volume 2, pp. 891–895. [CrossRef]
16. Brockwell, P.J.; Davis, R.A. *Introduction to Time Series and Forecasting*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2002. [CrossRef]
17. Fan, J.; Mu, D.; Liu, Y. Research on Network Traffic Prediction Model Based on Neural Network. In Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 28–30 September 2019; pp. 554–557. [CrossRef]
18. Lei, Y. Network Anomaly Traffic Detection Algorithm Based on SVM. In Proceedings of the 2017 International Conference on Robots & Intelligent System (ICRIS), Huai An City, China, 15–16 October 2017; pp. 217–220. [CrossRef]
19. Bereziński, P.; Jasiul, B.; Szpyrka, M. An Entropy-Based Network Anomaly Detection Method. *Entropy* **2015**, *17*, 2367–2408. [CrossRef]
20. Zhou, Y.; Li, J. Research of Network Traffic Anomaly Detection Model Based on Multilevel Autoregression. In Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 19–20 October 2019; pp. 380–384. [CrossRef]
21. Maniriho, P.; Niyigaba, E.; Bizimana, Z.; Twiringiyimana, V.; Mahoro, L.J.; Ahmad, T. Anomaly-based Intrusion Detection Approach for IoT Networks Using Machine Learning. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; pp. 303–308. [CrossRef]
22. Munther, A.; Alalousi, A.; Nizam, S.; Othman, R.R.; Anbar, M. Network traffic classification—A comparative study of two common decision tree methods: C4.5 and Random forest. In Proceedings of the 2014 2nd International Conference on Electronic Design (ICED), Penang, Malaysia, 19–21 August 2014; pp. 210–214. [CrossRef]

23. Alam, F.; Kashef, R.; Jaseemuddin, M. Enhancing The Performance of Network Traffic Classification Methods Using Efficient Feature Selection Models. In Proceedings of the 2021 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 15 April–15 May 2021; pp. 1–6. [CrossRef]
24. Marteau, P.F.; Soheily-Khah, S.; Béchet, N. Hybrid Isolation Forest–Application to Intrusion Detection. *arXiv* **2017**, arXiv:1705.03800.
25. Xiao, C.-H.; Su, C.; Bao, C.-X.; Li, X. Anomaly Detection in Network Management System Based on Isolation Forest. In Proceedings of the 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, China, 19–21 April 2018; pp. 56–60. [CrossRef]
26. Grewal, M.S. Kalman filtering. In *International Encyclopedia of Statistical Science*; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
27. Raj, S.; Singh, K.N.; Gupta, N.K.; Nigam, R.; Verma, B.; Karsoliya, S. High Accuracy of Hybrid IDS System using Evidence Theory and SVM ML Technique. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 1261–1264. [CrossRef]
28. Van Efferen, L.; Ali-Eldin, A.M. A multi-layer perceptron approach for flow-based anomaly detection. In Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2017; pp. 1–6. [CrossRef]
29. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 22–24 January 2018; pp. 108–116. [CrossRef]
30. Lim, H.K.; Kim, J.B.; Heo, J.S.; Kim, K.; Hong, Y.G.; Han, Y.H. Packet-based Network Traffic Classification Using Deep Learning. In Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Okinawa, Japan, 11–13 February 2019; pp. 46–51. [CrossRef]
31. Kim, T.; Suh, S.C.; Kim, H.; Kim, J.; Kim, J. An Encoding Technique for CNN-based Network Anomaly Detection. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2960–2965. [CrossRef]
32. University of New Brunswick–Intrusion Detection Evaluation Dataset (CIC-IDS2017). Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 5 December 2022).
33. Homoliak, I.; Malinka, K.; Hanacek, P. ASNM Datasets: A Collection of Network Attacks for Testing of Adversarial Classifiers and Intrusion Detectors. *IEEE Access* **2020**, *8*, 112427–112453. [CrossRef]
34. Homoliak, I.; Hanacek, P. ASNM Datasets: A Collection of Network Traffic Data for Testing of Adversarial Classifiers and Network Intrusion Detectors [Internet]. IEEE Dataport. 2019. Available online: https://ieee-dataport.org/open-access/asnm-datasets-collection-network-traffic-data-testing-adversarial-classifiers-and (accessed on 20 December 2022). [CrossRef]
35. United States Military Academy Westpoint–Cyber Research Center. Available online: https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets/ (accessed on 5 December 2022).
36. USMA Westpoint–Cyber Research Center–Cyber Defense Exercise. Available online: https://www.westpoint.edu/centers-and-research/cyber-research-center/cyber-defense-exercise/ (accessed on 5 December 2022).
37. Brno University of Technology–Security Laboratory Research Group–ASNM Datasets. Available online: https://www.fit.vutbr.cz/~ihomoliak/asnm/index.html (accessed on 5 December 2022).
38. BUT–Security LABoratory Research Group–ASNM-CDX-200 Dataset. Available online: https://www.fit.vutbr.cz/ihomoliak/asnm/resources/ASNM-CDX-2009.rarl (accessed on 5 December 2022).
39. USMA Westpoint–Cyber Research Center–CDX-2009 Dataset. Available online: https://drive.google.com/open?id=0B0u9Tg7udaAXaUFHRFpQWjR0dW8 (accessed on 5 December 2022).
40. Keras Documentation: LSTM Layer. Available online: https://keras.io/api/layers/recurrent_layers/lstm/ (accessed on 5 December 2022).
41. Keras Documentation: Dropout Layer. Available online: https://keras.io/api/layers/regularization_layers/dropout/ (accessed on 5 December 2022).
42. Keras Documentation: Dense Layer. Available online: https://keras.io/api/layers/core_layers/dense/ (accessed on 5 December 2022).
43. Kingma, D.P.; Ba, J. Available online: Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Keras Documentation: Regression Losses. Available online: https://keras.io/api/losses/regression_losses/ (accessed on 5 December 2022).
45. Keras Documentation: Probabilistic Losses. Available online: https://keras.io/api/losses/probabilistic_losses/ (accessed on 16 December 2022).
46. Scikit-Learn: Support Vector Machines. Available online: https://keras.io/api/losses/regression_losses/#mean_squared_error-function (accessed on 5 December 2022).
47. Scikit-Learn: Shuffle. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.utils.shuffle.html (accessed on 5 December 2022).

48. Kostadinov, S. Understanding GRU Networks. Available online: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be (accessed on 5 December 2022).
49. Limthong, K.; Tawsook, T. Network traffic anomaly detection using machine learning approaches. In Proceedings of the 2012 IEEE Network Operations and Management Symposium, Maui, HI, USA, 16–20 April 2012; pp. 542–545. [CrossRef]