sensors

MDPI

Article

Research of Machine Learning Algorithms for the Development of Intrusion Detection Systems in 5G Mobile Networks and Beyond

Azamat Imanbayev [1,2,*], Sakhybay Tynymbayev [3], Roman Odarchenko [4], Sergiy Gnatyuk [4], Rat Berdibayev [3], Alimzhan Baikenov [3] and Nargiz Kaniyeva [2]

[1] Faculty of Information Technology, Al-Farabi Kazakh National University, Almaty 050040, Kazakhstan
[2] School of Information Technology and Engineering, Kazakh-British Technical University, Almaty 050000, Kazakhstan
[3] Information Security Laboratory, Almaty University of Power Engineering and Telecommunications, Almaty 050013, Kazakhstan
[4] Department of Telecommunication and Radioelectronic Systems, National Aviation University, 03058 Kyiv, Ukraine
* Correspondence: imanbaevazamat@gmail.com

Abstract: The introduction of fifth generation mobile networks is underway all over the world which makes many people think about the security of the network from any hacking. Over the past few years, researchers from around the world have raised this issue intensively as new technologies seek to integrate into many areas of business and human infrastructure. This paper proposes to implement an IDS (Intrusion Detection System) machine learning approach into the 5G core architecture to serve as part of the security architecture. This paper gives a brief overview of intrusion detection datasets and compares machine learning and deep learning algorithms for intrusion detection. The models are built on the basis of two network data CICIDS2017 and CSE-CIC-IDS-2018. After testing, the ML and DL models are compared to find the best fit with a high level of accuracy. Gradient Boost emerged as the top method when we compared the best results based on metrics, displaying 99.3% for a secure dataset and 96.4% for attacks on the test set.

Keywords: intrusion detection system; 5G; dataset; machine learning; deep learning; cybersecurity

Citation: Imanbayev, A.; Tynymbayev, S.; Odarchenko, R.; Gnatyuk, S.; Berdibayev, R.; Baikenov, A.; Kaniyeva, N. Research of Machine Learning Algorithms for the Development of Intrusion Detection Systems in 5G Mobile Networks and Beyond. *Sensors* **2022**, *22*, 9957. https://doi.org/10.3390/s22249957

Academic Editor: Joel J. P. C. Rodrigues

Received: 9 November 2022
Accepted: 14 December 2022
Published: 17 December 2022

1. Introduction

Confidential information is often stored, transmitted, and processed in global networks. In this regard, the security of network systems is becoming increasingly important. The new fifth generation (5G) network architecture offers densely distributed storage, computing, and networking capabilities that provide more versatile services than previous generations, supporting a wider range of cases and applications. The scope of the network is being expanded by the increasing number of users (mobile devices, Internet of things), and it is expected that users will utilize 13 times more data in 2025 than now. This is forecasted based on an expected 21 billion devices connected to the Internet in 2025, in comparison to 7 billion [1] devices connected to the network today, which also implies an increase in hacker activity in the future. IoT is applied in various fields of research and is used in various applications such as Healthcare, Smart Grid, Transportation, Smart Home and Building, Smart Cities, Agriculture, Industry Automation, and the Military [2]. After 2030, wireless applications will require much higher data rates (up to 1 Tbps), extremely low end-to-end latency (<1 ms), and extremely high end-to-end reliability (99.99999%) [3]. However, such a great digital evolution is only possible with the next generation of 5G and 6G mobile networks. The migration to the cloud, virtualization, and the majority of network functions becoming software-assisted in 5G wireless networks and beyond is increasing the security risk of accessing core networks [4]. In the same way, the rapid development of technology

creates new security issues. As the main architectural pillar of 5G networks is to create programmable and configurable network components using software, it means one has to carefully check the software code before deployment in order to protect resources and user data. Hackers can exploit open vulnerabilities in the supply chain, which can lead to serious attacks which affect the entire network infrastructure. In the 5G network architecture and beyond, there has been a paradigm shift from the concept of dedicated network resources for dedicated network functions to the more dynamic virtualization, cloud organization, and orchestration of software-defined network functions from network resources [5]. All of these factors have the effect of increasing network insecurity and user data risk if malicious attacks are not detected in real time. In this context, machine learning and deep learning [6,7] are expected to play a vital role in the use of automated intelligence in 5G and other wireless networks. While 5G is well known for its cloud-based, microservices-based architecture, the next generation of the network, 6G is closely associated with intelligent orchestration and network management. Hence, the role of artificial intelligence (AI) in the 6G paradigm is of prime importance. AI is key to next generation 6G mobile networks, and ensuring its security is critical to realising the 6G concept. AI-enabled 5G and 6G mobile network security provides intelligent and reliable security solutions [8]. Nowadays, the concept of intelligent analytics needs to be implemented in all types of wireless networks, from local area networks to remote clouds. Network traffic prediction and estimation is a necessary part of network operations and management, such as congestion control, routing, resource allocation, and service level agreement management, as well as many other network responsibilities and functions [9]. Due to this chain of events, strong and effective security measures are required to create a safe and secure environment for users, but it is currently difficult to prevent attacks through passive security policies, firewalls, or other mechanisms. In addition, with the introduction of 5G technology, there will be security risks for older generations of mobile devices. Therefore, along with traditional security tools, such as firewalls, intrusion detection systems (IDS) are becoming increasingly important to help protect systems proactively. It is known that an IDS can collect traffic data (i.e., activity) and can analyse the received information. The intrusion detection system will be based on anomalies. This means that the system must go through a "familiarisation" period, during which it learns and remembers the current state of the infrastructure. What it learns becomes the benchmark against which the system will be guided in the future. In our case, we will train the system based on datasets. When monitoring a network, data is gathered from network packets [10–12]. Network attacks are carried either by embedding malicious code or analyzing network packets to gain information. Attacks can take place on either the server that processes all network transactions or on the system node that actually performs network activities. Actions can also be taken to exploit weaknesses in the system. Technologies such as machine learning and deep learning lead to improved intrusion detection systems (IDS) [13,14].

In this regard, in recent years researchers have explored the possibility of using artificial intelligence (AI) techniques to develop efficient IDS applications. In fact, machine learning methods have become one of the most promising tools for studying a wide range of complex issues, given the rapid growth of network traffic and security risks [14–17]. Research of the use of artificial intelligence technologies, in particular machine learning and deep learning, in network intrusion detection systems (NIDS), is a relevant topic but is still in its infancy, and there is still great scope for exploring these technologies in NIDS systems to effectively detect network intruders.

1.1. Background Analysis

Network security is one of the most discussed and important issues in a rapidly evolving society, as it affects the interests of many stakeholders. The rapid evolution of 5G mobile networks creates new risks, threats, and vulnerabilities in the system of which attackers can take advantage [18,19]. The ENISA report examined the challenges, vulnerabilities, and attacks on 5G networks [20], and the transition to 5G will involve

several phases, according to the 3GPP roadmap. One of these phases, 5G Non-Standalone, combines the use of the new 5G radio and LTE core network. As a result, these networks will inherit all the vulnerabilities of LTE networks. Studies show that LTE networks are vulnerable to denial of service (DoS) [21,22]. The best practice to defend against these attacks is to use virtual network security tools such as antivirus, virtual firewalls, or IDS/IPS to achieve a level of security comparable to traditional networks. In addition, machine learning (ML) enabled solutions can be used to detect attack traffic (e.g., DoS attacks) and distinguish it from normal traffic so that it can be handled accordingly [23]. Researchers and developers have a lot of work to do to ensure robust end-to-end security. Artificial intelligence (AI) and machine learning (ML) can play a vital role in the development and automation of next-generation mobile networks. The main advantage of the 5G network is its high data transfer rate, and it is more effective to use AI and machine learning to prevent and detect a wide range of threats from different points [19]. Fifth generation technologies, including Multi-access Edge Computing (MEC), SDN, NFV, and network slicing, are still relevant to 6G networks. Therefore, their associated security matters remain. For example, the most severe security concerns related to SDN include vulnerabilities on the SDN controller, interfaces, and SDN applications platforms. Security obstacles associated with NFV include attacks on virtual machines, hypervisors, and virtual network function (VNF) managers. Finally, MEC is vulnerable to physical risks, DDoS, and the enormously distributed structure of 6G systems [24]. The place of artificial intelligence in the 6G network architecture should also be taken into consideration. After all, artificial intelligence will appear in all parts of the network, including the borders of cells and, possibly, user devices. Under these conditions, the possibility of using these algorithms in the tasks of detecting and preventing cyber attacks becomes an obvious advantage.

Researchers have studied technologies, scenarios, and applications using artificial intelligence to secure 5G wireless networks. They have also come to the conclusion that AI can significantly increase the security of a distributed ad hoc configuration of the network infrastructure that provides various network functions. However, more thorough research is required before AI fully takes over the digital automation of mobile networks [25].

Only artificial intelligence (AI) tools, especially machine learning (ML) and deep learning (DL) [26–29], can handle the real-time analysis of the huge volumes of data traffic that is generated in fifth generation networks. Despite the great possibilities of creating self-managed networks with the help of AI, attacks on algorithms can lead to significant performance degradation and network failures [30].

1.2. Related Works

This paper explores the potential of machine learning in IDS to secure 5G networks. It is anticipated that AI will be a key enabler of 5G and other networks [31]. In the past it has become obvious that artificial intelligence (AI)-based techniques play a prominent role in the ensemble development for intrusion detection and have many benefits over other techniques [32]. Here, an updated general review of ensembles and their taxonomies has been presented. The paper also presents the updated review of various AI-based ensembles for IDS (in particular) during the last decade. Various IDS systems exist currently and the authors have presented an in-depth review of intrusion detection systems (IDS) for the IoT from 2015 to 2019 [4]. In [33], various AI based techniques have been reviewed focusing on the development of IDS. A lot of studies were devoted to the investigation and comprehensive analysis of different approaches for detecting different attacks in different conditions depending on the available data collected [34–37]. A framework to build and operate AI-based intrusion detection for in-situ monitoring was described and analysed in detail in [38].

A large number of papers analyse different IDS datasets and one study provides information on the latest CIC IDS 2017 dataset. The publication discusses the application of IDS as well as potential future research directions [14]. In the same way, the researchers conduct experiments with two reference datasets, namely NSL-KDD and CI-

CIDS2017 [6]. Another article discusses CIC-IDS-2017 and CSE-CIC-IDS-2018, as well as a review of the ML and DM algorithms used for IDS. These are the most recent datasets which provide characteristics of network attacks, which include new types of attacks [7]. Deep learning is already widely used to solve the detection problems of various network attacks [39–41]. In [42], the Deep Neural Network on NSL-KDD dataset was researched for effective attack detection.

The performance of IDS is one of the key factors. The researchers [43] in this study focused on improving the performance of DNN-based IDS by providing a unique feature selection method that combines statistical significance using standard deviation and the difference of mean and median. The effectiveness of the proposed approach is evaluated using three intrusion detection datasets: NSL-KDD, UNSW NB-15, and CIC-IDS-2017.

In addition, the criteria that are set for the datasets play an important role. Markus Ring [44], for example, discusses common aspects of dataset descriptions and divides them into five categories. This study provides a focused assessment of datasets for network-based intrusion detection, as well as specifics on the underlying packet- and flow-based network data. The report identifies 15 alternative parameters for assessing individual dataset applicability for specific evaluation scenarios.

Furthermore, it was shown that one of the best solutions for monitoring and detecting threats in 5G networks is to use AI-based IDS trained on big data [45]. Similarly, the main security problem is the development of a methodology for detecting malicious activity, due to the fact that it is necessary to update the database with malicious traffic for AI training [46]. In the literature, many researchers have proposed different ensembles by considering different combination methods, training datasets, base classifiers, and many other factors [47,48]. However, the task to identify the most correct usage of datasets remains open. Researchers have proposed a methodology for integrating intrusion detection systems into the standard 5G architecture [49].

However, no cases on network security have shown applications of various deep learning algorithms in real-time services beyond experimental conditions in 5G networks.

We propose a DNN-based intrusion detection and classification system, which takes into account statistical indicators to evaluate the performance of models. After preprocessing the data, we performed feature engineering to select and transform features that can be used to build the model. Two public datasets are used for implementation, such as CICIDS2017 and CSE-CIC-IDS2018.

1.3. Problem Statement

The architecture of traditional networks has not changed for decades, which brought many problems and highlighted several security issues, and the development of 5G networks has raised even more concerns about the future structure of the Internet.

To advance with this new technology, the use of software-defined networks is essential to begin the successful deployment and implementation of a powerful wireless world [50]. Several researchers have repeatedly said that the SDN architecture has many advantages as it provides many solutions to the problems of legacy network infrastructure, which has attracted the attention and interest of scientists [51,52]. Thus, it is seen as a new software-based network architecture that could offer significant benefits for 5G networks. The most notable features of this network are that it is low cost, flexible, expandable, and it increases the size of its infrastructure without the complexity of a traditional network. This architecture consists of three main layers (control plane, controller, and application plane), and all operations in this architecture are controlled by the controller [53]. Since this element is considered to be the brain of the network, it is completely isolated from the network and if attackers attack it, it will lead to the downfall of the entire network. Accordingly, the controller is the most malicious part and the most vulnerable to attacks. In response to this threat, the need to develop an intrusion detection system (IDS) has emerged and grown. This is because it constantly monitors the network and creates a traffic

pattern that enables it to detect behavior or traffic patterns that deviate from the normal pattern [54].

In this paper, we offer deep learning technologies as they can quickly and accurately identify a wide range of attacks. However, conventional types of Machine Learning (ML) algorithms and many types of Deep Learning (DL) algorithms are initially used to evaluate them based on various criteria (Accuracy, F-score, Recall, Precision, etc.) [55,56]. To train our classifiers based on strongly related features, we used feature selection approaches. This data can then serve as a basis for new researchers willing to start exploring this promising area. The main contributions of this article are listed below:

- Overview of available datasets for building smart LEDs. A comparative analysis between them is also presented, highlighting their metadata, types of attacks, format, etc.
- An overview is provided on the work of similar topics on the application of ML and DL in NIDS, which have previously been explored by other researchers.
- Comparison of the performance of Logistic Regression (LR), Gradient Boosting, Random Forest (RF), Autoencoder, and Deep Neural Network (DNN) with hyperparameter search.
- Unresolved issues in the development of NIDS based on machine and deep learning are highlighted.

2. Proposed Methodology and Model Classifiers

Implementing an intelligent system to detect intrusions into the core network can be achieved through software-defined security. This is because the two main components of a 5G network, RAN and the core network, are virtualized and are fully software-defined. Therefore, it is possible to go in one direction and create an automated security system. The idea is that copies of the traffic from the backhaul connection and the core network are sent to SDS for analysis [57]. It should be noted that the copies of the traffic do not affect performance in any way, while the network is being analysed. However, before it can be determined whether or not the traffic is anomalous, the data must be pre-processed to make it more readable and easier to use for machine learning or deep learning models. The anomalies are then analysed with the appropriate algorithms and the results are sent to the Policy Manager database. The results are then forwarded to the VNF Manager, which updates the module IDS. Here, the model processing time plays the most important role in presenting the final results. In other words, this helps determine when the template needs to be run so that the module policies are up to date. Thanks to this technique, it is possible to automate the detection, the update of the attack database, and the actions taken to defend the network against intruders.

Figure 1 shows a possible implementation of the IDS module in the 5th generation mobile network.

One of the most important factors in the development of ML-DL-based IDS in SDN is the appropriate selection of datasets. There is an obvious lack of studies on datasets used in ML-based IDS-SDN research. Nevertheless, relatively few of them have applicable types of attacks and properties that could help in implementing models in practice. In this part of the section, we address the main difficulties that researchers encounter in developing an intelligent intrusion detection system.

First of all, it is very difficult to collect reliable data for research. This is because technology changes several times a year, which increases security threats and updates the list of new attacks. As a result, datasets quickly lose their importance and value in the cybersecurity society. The second problem is the integrity of the dataset. This means that researchers need to include not only CSV files, but also audit logs and raw data from the network. Audit logs can be used to find important information about cyber-attacks, and raw data improves threat detection. The next point to consider is the types of attacks. As technology advances, new types of attacks emerge as hackers adapt their attacks to current systems or software, creating a vicious cycle. In this case, two methods come into question:

the use of new datasets or the dataset generator, which adapts like a hacker and creates corresponding attacks.

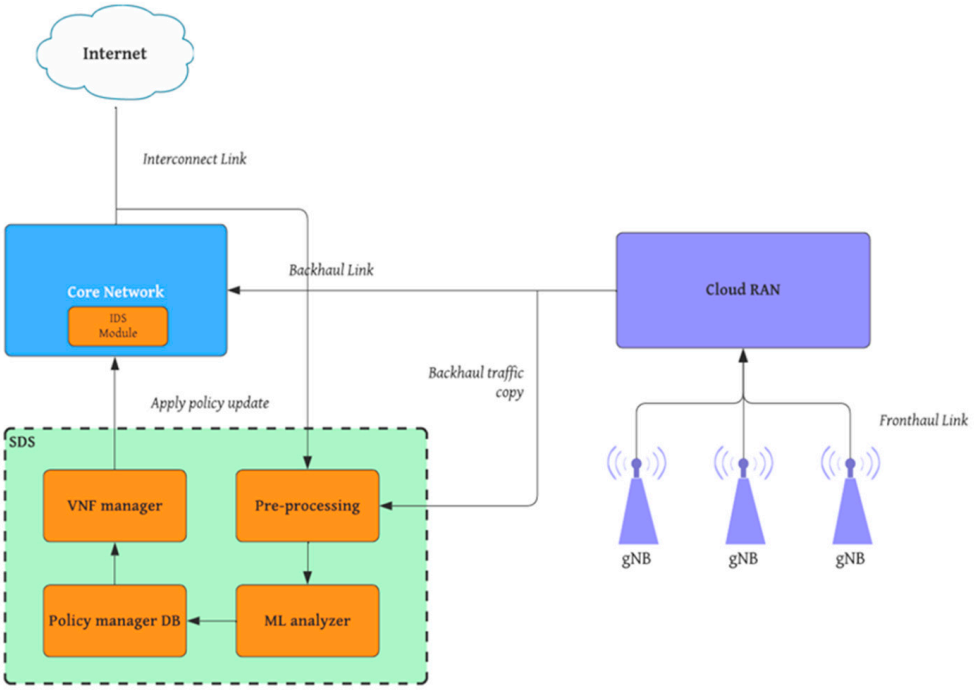


Figure 1. 5G network with IDS module system.

In addition, the generated datasets must be as realistic as possible if stakeholders are to use the model in production. In other words, they must contain normal traffic from various end-user workstations and servers. Otherwise, the trained model may not be suitable for a particular network. It should be noted that data privacy is also considered in the dataset. Although the most trusted data sources are the providers' mobile networks, they are not always willing to share their data (audit log or network logs) as this violates privacy policies. Therefore, researchers do not train and test their model with real network traffic data, but usually use popular datasets where the data is modeled [6,14,40,42].

The need for labeling is also high [58]. This is true whether it is supervised or unsupervised learning, as labeling is required to calculate the accuracy of the algorithm used. In fact, experts use cyberspace to collect secure network activity before using the data to attack network traffic. Therefore, they set up normal traffic first and then attack. Some experts insert attacks into normal traffic, while others do a manual tagging, which makes the latter process more laborious. Finally, the dataset needs to be widely accepted by the research community in order for the scientific work to be appreciated. Without this support, the dataset can only be used in a few research projects.

As written in the related works, modern scientists have done a lot of research on dataset analysis for IDS. The algorithms used for IDS are implemented on the DARPA dataset [59], KDD CUP 99 [60], NSL-KDD [47], or UNSW-NB15 [61] in which the network instances are grouped as training and test sets. The CIC-IDS-2017 and CSE-CIC-IDS2018 datasets present a new spectrum of generated attacks based on real network traffic characteristics. Table A1 in Appendix A gives details.

5G networks have become the backbone of the Internet of the future. At the same time, it is obvious that the functioning of this type of network will work according to the principle of the all-over-IP architecture. Under these conditions, it is clear that in new networks, along with a large number of new cyber attacks, there will remain no fewer well-known and quite advanced criminal network attacks aimed at vulnerabilities in network architectures using the TCP/IP protocol stack. Therefore, if we look at the 5G network

from an Internet-based perspective, the set of parameters included in the CICIDS2017 and CSE-CIC-IDS2018 datasets can be used to train AI-based IPS/IDS to detect common network attacks. Moreover, many researchers [62–64] have considered the prospects of 5G and network architecture based on the SDN principle.

The authors in [7,17] present an exhaustive survey on IDS based on CICIDS2017 and CICIDS-2018 datasets. They examined numerous research papers and compared their performances based on their ML models, computing environments, and several performance parameter scores such as accuracy, precision, recall, the area under the curve, etc. The CICIDS2017 and CSE-CIC-IDS-2018 datasets can be a convincing dataset to evaluate ML-based IDS in the 5G network.

Datasets such as CICIDS2017 [65] and CSE-CIC-IDS2018 [66] have been considered in this study. In the first case, the authors studied the model using the dataset, and in the second case, they evaluated the performance of the model. There are many other datasets available on the Internet to monitor network traffic, but some of them are outdated, inflexible, and have duplicate credentials. Figure 2 describes the features of the two selected datasets.

No.	Feature	No.	Feature	No.	Feature
1	Flow ID	29	Fwd IAT Std	57	ECE Flag Count
2	Source IP	30	Fwd IAT Max	58	Down/Up Ratio
3	Source Port	31	Fwd IAT Min	59	Average Packet Size
4	Destination IP	32	Bwd IAT Total	60	Avg Fwd Segment Size
5	Destination Port	33	Bwd IAT Mean	61	Avg Bwd Segment Size
6	Protocol	34	Bwd IAT Std	62	Fwd Avg Bytes/Bulk
7	Time stamp	35	Bwd IAT Max	63	Fwd Avg Packets/Bulk
8	Flow Duration	36	Bwd IAT Min	64	Fwd Avg Bulk Rate
9	Total Fwd Packets	37	Fwd PSH Flags	65	Bwd Avg Bytes/Bulk
10	Total Backward Packets	38	Bwd PSH Flags	66	Bwd Avg Packets/Bulk
11	Total Length of Fwd Pck	39	Fwd URG Flags	67	Bwd Avg Bulk Rate
12	Total Length of Bwd Pck	40	Bwd URG Flags	68	Subflow Fwd Packets
13	Fwd Packet Length Max	41	Fwd Header Length	69	Subflow Fwd Bytes
14	Fwd Packet Length Min	42	Bwd Header Length	70	Subflow Bwd Packets
15	Fwd Pck Length Mean	43	Fwd Packets/s	71	Subflow Bwd Bytes
16	Fwd Packet Length Std	44	Bwd Packets/s	72	Init_Win_bytes_fwd
17	Bwd Packet Length Max	45	Min Packet Length	73	Act_data_pkt_fwd
18	Bwd Packet Length Min	46	Max Packet Length	74	Min_seg_size_fwd
19	Bwd Packet Length Mean	47	Packet Length Mean	75	Active Mean
20	Bwd Packet Length Std	48	Packet Length Std	76	Active Std
21	Flow Bytes/s	49	Packet Len. Variance	77	Active Max
22	Flow Packets/s	50	FIN Flag Count	78	Active Min
23	Flow IAT Mean	51	SYN Flag Count	79	Idle Mean
24	Flow IAT Std	52	RST Flag Count	80	Idle Packet
25	Flow IAT Max	53	PSH Flag Count	81	Idle Std
26	Flow IAT Min	54	ACK Flag Count	82	Idle Max
27	Fwd IAT Total	55	URG Flag Count	83	Idle Min
28	Fwd IAT Mean	56	CWE Flag Count	84	Label

Figure 2. CICIDS2017 and CSE-CIC-IDS2018 network traffic features [67].

Complete form of the CICIDS2017 dataset that contains 3,119,345 instances and 83 attributes containing 15 class labels (1 normal + 14 attack labels). The prevalence of the majority class (Benign) is 83.34% and that of the minority class is 0.00039% (Heartbleed). CSE-CIC-IDS2018 was generated from a significantly bigger network of simulated client-targets and attack machines [7], yielding a dataset of 16,233,002 instances acquired from 10 days of network activity. Approximately 17% of the occurrences are assault traffic.

Evaluation Metrics

To evaluate the developed IDS models for the SDN based 5G network and then compare them, indicators such as accuracy, precision, recall, and F1-score are used [68].

All these metrics are determined by the following characteristics:

1. True Positive (TP): The malicious flow is classified as 'malicious' by the model and the result is a true positive.
2. False Positive (FP): The malicious flow is classified as 'benign' by the model, the result is a false positive.
3. True Negative (TN): The benign flow is classified as 'benign ' by the model and the result is True Negative.
4. False Negative (FN): The benign flow is classified as 'malicious' by the model, resulting in a false negative result.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall\ (DT) = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

3. Experiments and Result

In this section, we show the machine and the deep learning algorithms they used for their work.

3.1. Data cleaning

To ensure that their data is prepared for the analysis phase, the company will benefit greatly from data cleansing, which improves data quality [69]. The process of preparing the data for analysis before designing the model should be done by filtering out unnecessary or misleading information (e.g., data cleaning). Datasets are usually collected and merged into smaller files, which could lead to some duplicates and unwanted items. It is worth noting that an incorrect data collection method can lead to the misrepresentation of data and a decrease in the accuracy of models. In addition, models trained on the wrong datasets may perceive the noise as valuable information, and when it comes to training, it will show a good result. However, when cleaned datasets are input into it, unsuccessful results are displayed.

Consequently, the following manipulations were carried out on the CSE-CIC-IDS2018 dataset:

1. Removing invalid lines
2. Removing invalid values
3. Cleanup script

In CSE-CIC-IDS2018, the dataset stores ten separate CSV files, each containing recorded network traffic for one day of operation, named after the day the traffic was recorded. Therefore, one file (i.e., 'Thursday-01-03-2018_TrafficForML_CICFlowMeter.csv') is loaded and analysed for the initial analysis of the dataset.

As for the first step, when querying information on columns, the first problem encountered is that pandas outputs all columns as object columns, not numeric columns, which is fine for the most of them. To understand the reason why columns are interpreted as objects, the sub-columns are analysed to reveal individual values (Figure 3).

A distinct value indicates that the column name exists as a value in the dataset. A visual inspection of the input file reveals multiple occurrences of the title in the file combined with the original data path. This indicates that the file was created by merging multiple CSV files, with repeating titles. To resolve this issue, all header columns are removed from the data frame.

```
df['FIN Flag Cnt'].value_counts()

0                268629
0                 60520
1                  1707
1                   244
FIN Flag Cnt         25
Name: FIN Flag Cnt, dtype: int64
```

```
df['Protocol'].value_counts()

6            170066
17            95674
6             42833
17            15378
0              4596
0              2553
Protocol         25
Name: Protocol, dtype: int64
```

Figure 3. info() method.

Closer visual inspection of the file reveals the presence of the infinity chain in several rows of this column. The pandas read_csv() method cannot correctly parse this value because it only recognizes inf/-inf strings as a valid representation of infinity. To solve this problem, all occurrences of infinity are replaced by the string 'inf'.

After correcting the data in one file, the same must be done in the remaining nine files. Therefore, a script was written so that all datasets go through data cleaning:

- Deleting duplicate headers entered as dataset rows.
- Replacing occurrences of 'Infinity' with 'inf'.
- Renaming columns to remove spaces and characters without words.

The script (Appendix B) processes all files in the dataset and saves the output file with a name that describes the type of streaming attack in the file.

3.2. Exploratory Data Analysis

The next step is exploratory data analysis (EDA) [70–72]. This approach is useful for visualizing data and finding answers to a specific task.

In this work, the authors discovered the amount of safe and malicious network flows that the dataset contains and the amount of network streams that each type of attack contains. A strong correlation between certain features was examined to understand which features are worth paying attention to.

After conducting an exploratory analysis of the data, the following properties were revealed:

- The dataset is not balanced, so safe network traffic far outweighs malicious traffic (Figure 4).
- Another problem with datasets is the types of attacks, some of which are poorly specified (Figure 5). Therefore, it is difficult to recognize these attacks when training multi-class classifiers.
- When a correlation was established between features in the dataset, a strong correlation was found, suggesting the idea that the dataset contains redundant data. This must be taken into account when selecting and extracting features.
- Several features were then analysed to find predictors for the binary classifications. Due to numerous features, the visual identification of potential features is not possible. In order to identify features with high predictability, it is proposed to use a functional selection and extraction process such as PCA.

In general, the main predictors for binary classification were found:

- fwd_seg_size_min
- bwd_pkts_s
- ack_flag_cnt
- fwd_seg_size_min
- bwd_pkt_len_min

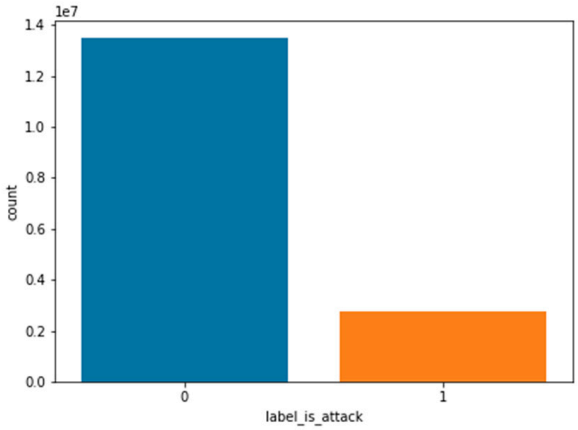


Figure 4. Number of benign and malicious flows.

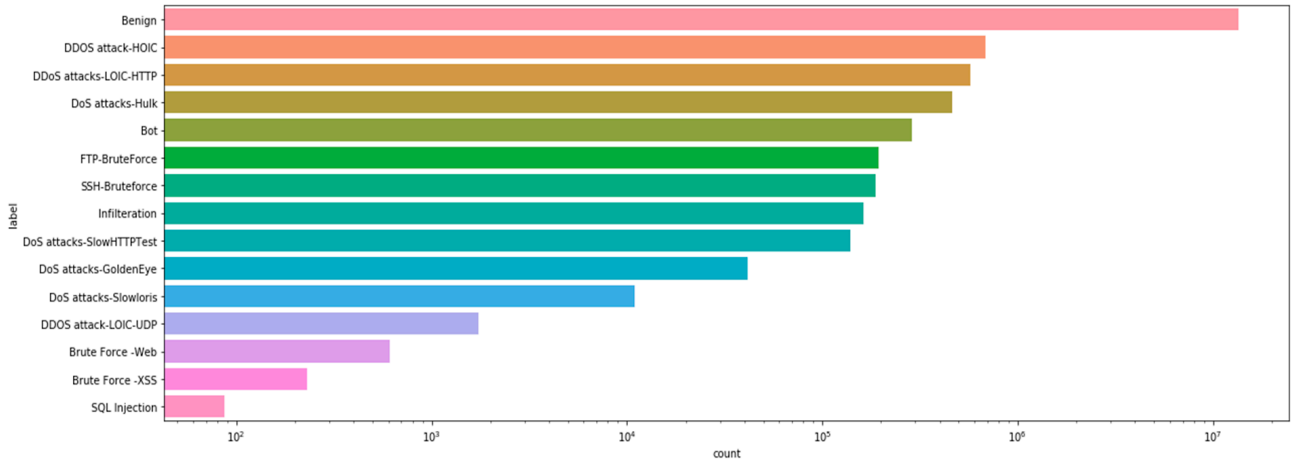


Figure 5. Number of flows/attack type.

3.3. Building ML Model Prototype

A formal branch of research called "machine learning" focuses on the theory, effectiveness, and characteristics of learning algorithms and systems. It is a highly interdisciplinary field based on concepts from numerous disciplines of science, engineering, and mathematics, including artificial intelligence, optimization theory, information theory, statistics, cognitive science, optimum control, and many others [73–76]. Machine learning has nearly all scientific fields covered thanks to a wide range of applications, which has had a significant impact on both research and society [77]. Numerous issues, including those related to recommendation engines, recognition systems, computer science and data mining, and autonomous control systems, have been resolved using it [78].

Since 5G generates more data at a faster rate than previous generations, telcos must be able to collect and analyse it at scale. By investing resources in the development of standards and a seamless framework that enables data governance, data integration, a modern data architecture that allows data to be accessed regardless of where it resides, and the ability to perform analytics on the database at any scale, current and future needs for 5G analytics can be met.

3.4. Experiment—1

This step used various types of existing algorithms to create binary classifications [79,80] that can distinguish between secure network traffic and malicious traffic based on the CIC-IDS-2018 dataset, for example:

- Logistic regression [81]

- Random forest [82]
- Gradient Boosting [83]

Timestamp and dst_port features were not included in the model, so the attack can be recognized regardless of the time and port of the target performing the attack. To do this, both features were removed from the dataset. After detecting the high correlation, the next step was to delete these properties. To confirm these characteristics, hierarchical clustering was performed on the Spearman rank-order correlations [84]. After choosing a threshold, an attribute from each cluster was stored in the dataset. In the end, the remaining number of features was 31. The reason for removing these features was that they did not affect the predictability of the model in any way, but rather caused noise (Figure 6 shows a correlation heatmap after removing highly correlated features).

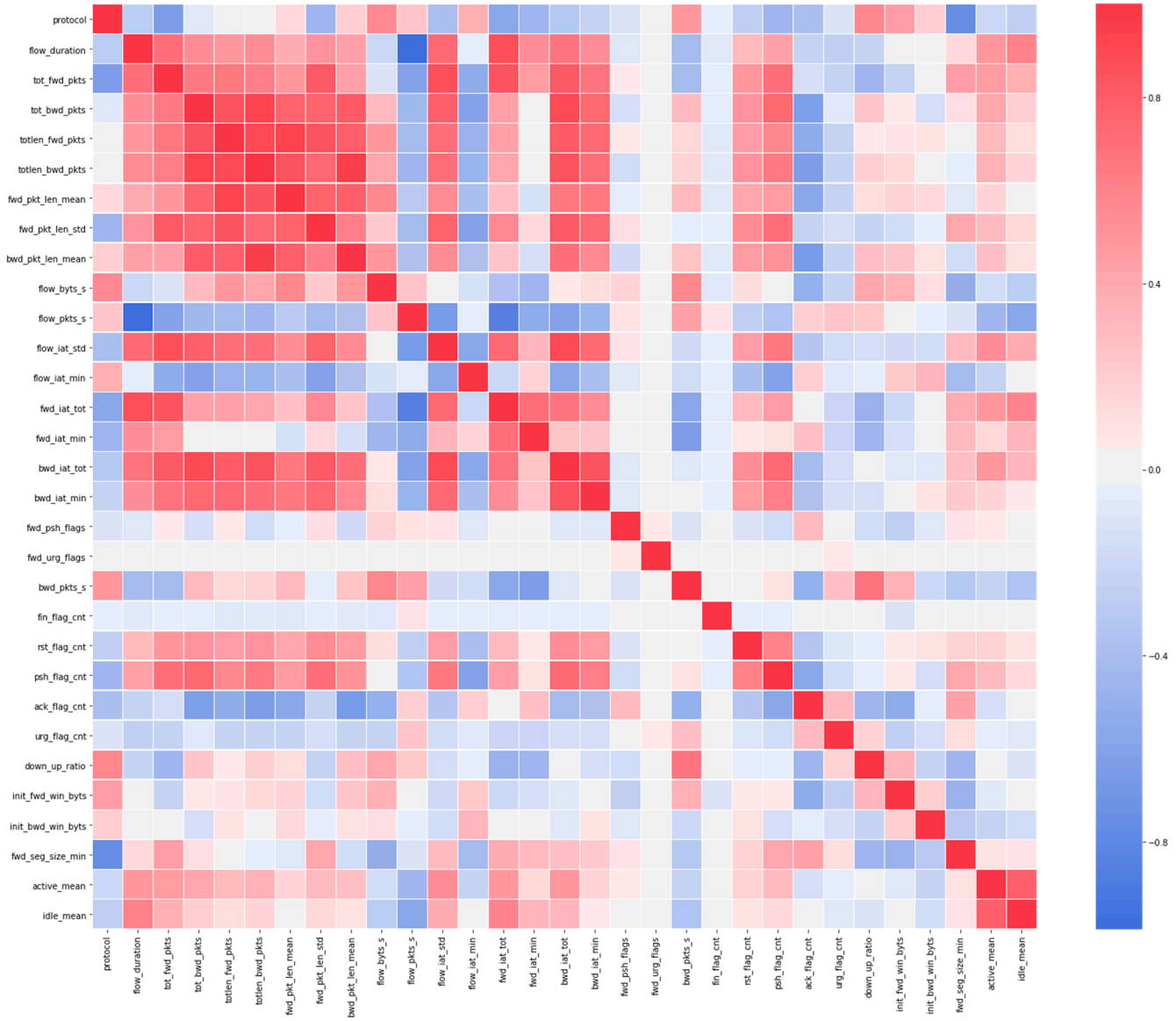


Figure 6. Correlation heatmap after the removal of highly correlated features.

After that, there is a step of dividing the dataset into training, evaluation, and testing with ratios of 0.8, 0.1, and 0.1, respectively. Then, it turns out that the dataset is highly unbalanced: class 0–benign makes up about 83% of all samples. For this reason, two metrics were used to evaluate the classifier, namely Recall and Precision.

Since the goal of the classifier is to identify as many attacks as possible, the first indicator was used as the main metric. The second one (i.e., Precision) was used as a secondary classifier, since the number of false positives should be kept to a minimum. This metric must have a value above the 0.95 threshold in order to have a maximum of 5% false positives.

When developing a machine learning model for any project, it is best to start with a baseline model. It is a Dummy model [85] that consistently predicts the most commonly encountered class. In the IDS method, the base model defines the system's normal or expected actions and compares all network actions or traffic to this base model.

The first experiment was the application of logistic regression algorithm, which are used to observe discrete classes. To use logistic regression, the predictors are scaled using the standard scaler. Although this is a popular algorithm, it has the disadvantage of being sensitive to outliers. In the end, the algorithm surpassed the baseline, showing a weighted recall of 0.88 and a precision of 0.87, which is not enough for practical application. The next algorithm to be evaluated is the random forest classifier implementation from scikit-learn. The Random Forest classifier performs very well with a recall of 0.99 and a precision of 0.99. This group classification (Random Forest) works better than other traditional classifiers to effectively classify attacks even with default values. In the latest algorithm, gradient boosting using the CatBoost library [86] was used. This algorithm also makes predictions based on an ensemble of other algorithms. Its main difference with a random forest is the sequence of tree creation, while in the previous algorithm a decision tree was created for each sample. In the research, we applied a grid search using cross-validation on various hyperparameters performed to determine the optimal parameters (Table 1).

Table 1. Results for training dataset.

No. of Selected Features: 31		Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Technique	**Accuracy**	**Class 0**				**Class 1**			
Logistic Regression	0.88	0.90	0.95	0.93	10,787,766	0.68	0.49	0.57	2,198,588
Random Forest	0.99	1.00	0.99	0.99	10,787,766	0.96	0.98	0.97	2,198,588
Gradient Boosting	0.99	0.99	0.99	0.99	107,877,66	0.97	0.96	0.96	2,198,588

Table 2 shows the evaluation metrics of four algorithms to select a single model and continue the experiment. From the table, the Gradient Boost algorithm slightly outperformed Random Forest, so it passes to the next evaluation stage, namely model testing. The final score shows very good performance, with a recall of 0.99 and a precision of 0.99.

Table 2. Model selection.

Model	Precision Benign (0)	Recall Benign (0)	F1-Score	Avg Precision	Recall Attack (1)	Precision Attack (1)
Logistic Regression	0.86	0.88	0.87	0.422	0.49	0.68
Random Forest	0.99	0.99	0.99	0.925	0.95	0.96
Gradient Boosting	0.99	0.99	0.99	0.933	0.96	0.97

However, Figure 7 shows the misclassifications in the test dataset, which demonstrates that "Infiltration" attacks are often misclassified. Furthermore, minority attack classes are

often misclassified. To improve performance, synthetic minority resampling can be applied to the training dataset for these classes.

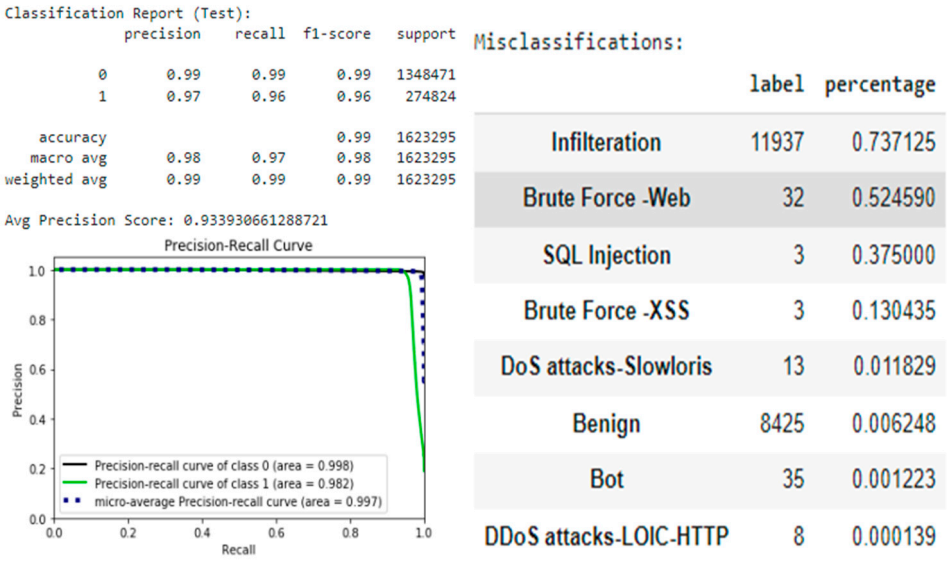


Figure 7. Gradient Boost performance on test dataset.

To ensure that the estimator has the same good performance as shown in the test dataset, additional tests were performed on the CIC-IDS-2017 dataset, which contains the same attack scenarios but is recorded in a different network environment. However, the estimator performed very poorly with data recorded in a different network environment (Figure 8), showing a recall of 0.82 and a precision of 0.80. Moreover, the estimator had an attack recall of only 0.26, which is not sufficient for real-world networks. This result suggests that data from one network environment does not generalize well enough to another network environment.

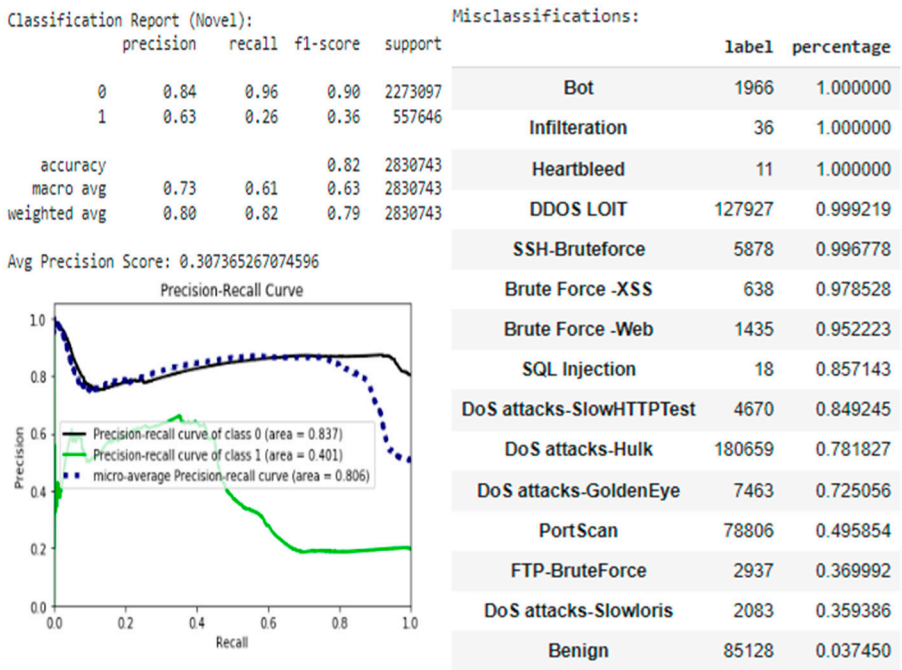


Figure 8. Testing on novel data.

Therefore, to analyse the problem, the values of the features of the model are calculated. The most important features will be used to compare data from two datasets. The Kolmogorov–Smirnov plots and statistics [87] assume that all features come from different distributions in both datasets, which poses a problem for the estimator because it assumes that the training, test, and real data come from the same distribution. Hence, the Kolmogorov–Smirnov statistic was performed for all features of the datasets. After removing the no variance feature, there are only two features coming from the same distribution in both fwd_urg_flags and cwe_flag_count datasets, both of which are not good predictors. This shows that data from different network environments is distributed differently.

In order to create an estimator that summarizes data well from different network environments, the estimator is created using the combined CIC-IDS-2017 and CIC-IDS-2018 datasets. Both datasets contain attack classes with a small number of cases. To get a higher detection rate for these attacks, Synthetic Minority Oversampling is used to increase the occurrences of these classes to 100,000. For a combined estimation, the gradient boosting model is trained using grid search to find the best set of hyperparameters.

It should be noted that the combined estimator shows promising performance on the test dataset with a high recall of 0.99, a precision of 0.99, and an attack detection rate (recall class 1) of 0.96 (see Figure 9 for combined estimator results).

```
Classification Report (Train):
              precision    recall  f1-score   support

           0       0.99      0.99      0.99  12606243
           1       0.98      0.97      0.97   3285956

    accuracy                           0.99  15892199
   macro avg       0.99      0.98      0.98  15892199
weighted avg       0.99      0.99      0.99  15892199

Avg Precision Score: 0.955579742683623


Classification Report (Test):
              precision    recall  f1-score   support

           0       0.99      0.99      0.99   3151562
           1       0.97      0.96      0.97    661176

    accuracy                           0.99   3812738
   macro avg       0.98      0.98      0.98   3812738
weighted avg       0.99      0.99      0.99   3812738

Avg Precision Score: 0.9425657531936107
```

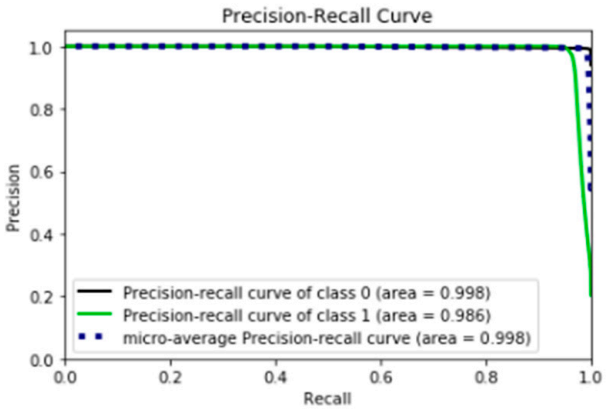


Figure 9. Combined estimator.

The following list (see Figure 10) shows the feature importances of the combined estimator calculated with permutation importance.

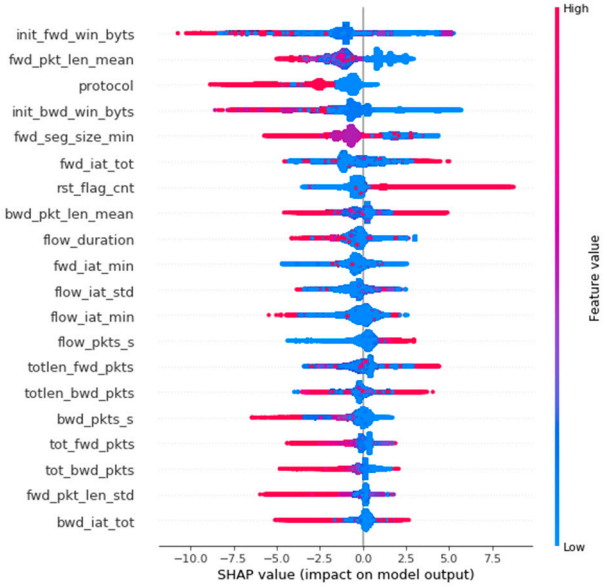


Figure 10. Feature importance.

According to the SHAP analysis [88], the following features have the greatest impact on the model assumptions:

1. init_fwd_win_byts: Number of bytes sent in the initial window in the forward direction.
2. fwd_pkt_len_mean: Mean size of packet in forward direction.
3. protocol: Protocol.
4. init_bwd_win_byts: Number of bytes sent in the initial window in the backward direction.
5. fwd_seg_size_min: Minimum segment size observed in the forward direction.

While the performance of the combined estimator is convincing, it can be assumed that the estimator will not generalize well across different network environments due to observed differences in distributions. The statistical characteristics recorded in the individual datasets appear to be highly dependent on the network topology and the configurations of the host and client machines on the network.

To solve this problem, the following suggestions are offered:

- The estimator should train on more diverse data coming from different network environments.
- The evaluator must be trained with data obtained from the network environment in which it will be deployed.

The second option seems to be more promising since it is very difficult to obtain high-quality datasets on real network attacks. The fact that data must be collected in the target environment can be alleviated by collecting only secure network traffic and using an anomaly detection approach to detect network attacks.

3.5. Experiment—2

In the second part of the experiment, the authors used an unsupervised learning approach to create a binary classifier based on the ideas of representation learning and anomaly detection. The idea was that several deep learning models were trained on benign data from the CIC-IDS-2018 dataset in order to learn the meaningful representation of these benign data. With this approach, there was a chance to create a model capable of classifying network traffic as safe or malicious, based on the notion of similarity or dissimilarity of the traffic to the data on which the model was trained. The rationale for using unsupervised learning is that useful data is usually easier to obtain and therefore can be provided in larger volumes than malicious data. For this, the authors decided to apply the autoencoder architecture (neural network).

It is known that an autoencoder learns to reconstruct given inputs by initially encoding the input features as dense representations and then decoding the dense representations to reconstruct the original input. Using this approach, the model must learn an income identification function.

As mentioned above, given the use case of network traffic classification, the model is only trained on good data. A secure and malicious data validation set is used to determine a decision boundary based on both types of traffic and the reconstruction errors.

On inference, the sample is fed into the autoencoder, the reconstruction error is measured, and then the sample is classified as malicious if the reconstruction error exceeds a predetermined decision boundary.

For all this, three variants of the autoencoder architecture are taken into account (Undercomplete, Stacked, Denoising).

Summarising the results of the experiments using anomaly detection, it can be seen that the performance of the resulting model is insufficient for real use and significantly worse than the performance of machine learning models created in previous experiments (see Figure 11 to compare performance results of autoencoders).

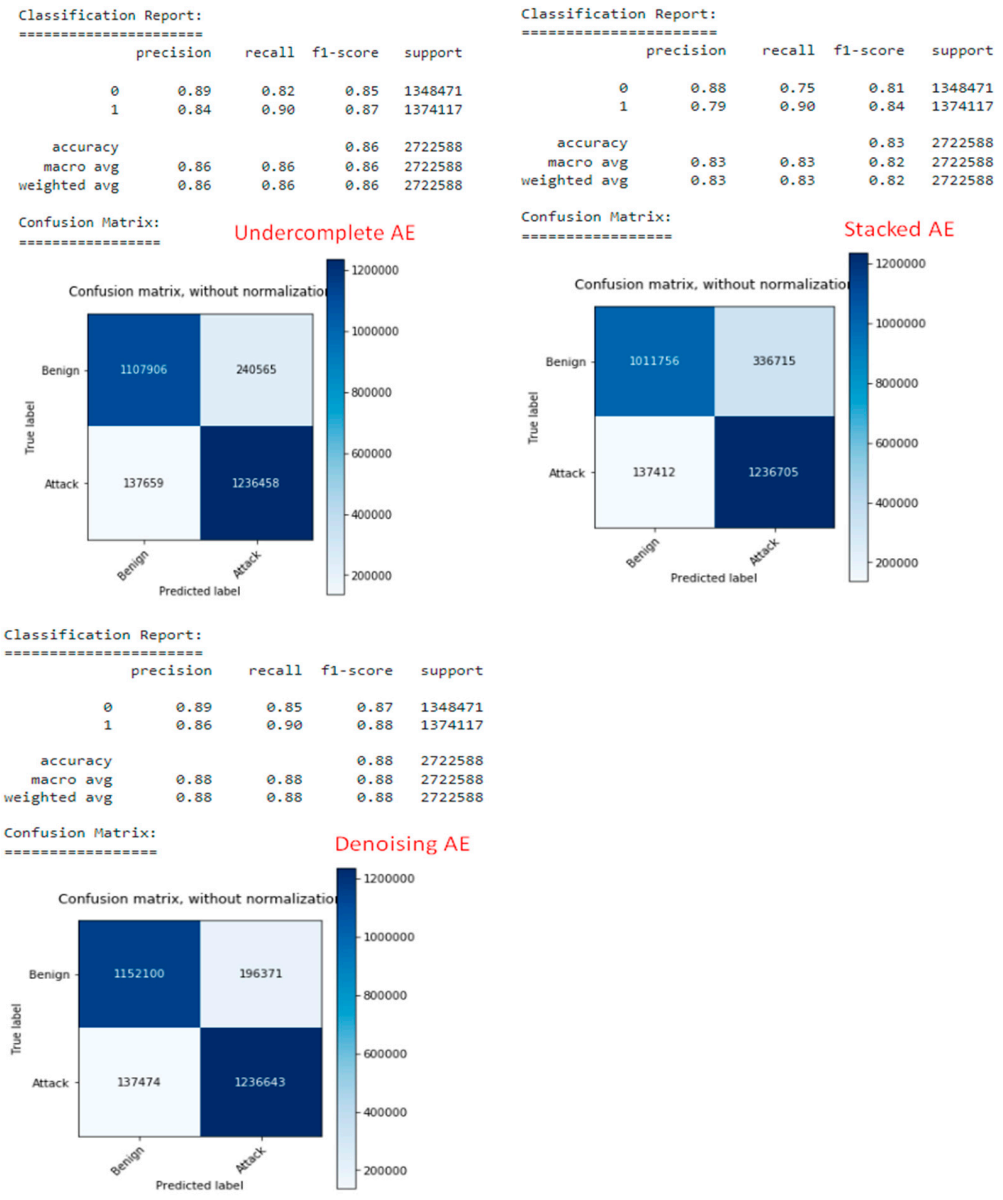


Figure 11. Performance of Undercomplete, Stacked, Denoising Autoencoders.

Moreover, the predictions of this estimator are very sensitive to the chosen value of the decision boundary, which can be reliably determined only if there is a sufficient amount of malicious data. This circumstance somewhat reduces the usefulness of this approach, since the biggest advantage of this method is the assumption that only secure data is needed to create an evaluator, and the collection of malicious data is not required or strictly limited.

The demonstrated approach can be useful in situations where malicious training data is only available in small amounts or is not available at all. If there is no malicious training data, the choice of the decision boundary can be made by determining a reasonable confidence interval taking into account the distribution of safe samples and adjusting the boundary when new data arrives.

3.6. Experiment—3

In this experiment, the authors used a supervised learning approach to create a binary classifier capable of distinguishing between safe and malicious network traffic. Several deep neural network models were selected using network traffic data taken from the CIC-IDS-2018 dataset and their respective characteristics were evaluated.

The values of the dataset target variable are grouped into two classes: benign and attacking, while the attack class includes all types of malicious network traffic. Since the dataset is highly unbalanced and contains 83% safe and only 27% malicious samples, this class imbalance is taken into account during training.

In the first part, a simple deep network was trained using two different approaches. The first approach does not take class imbalance into account, while the second approach uses class weights to weight the underrepresented sample loss more heavily during training. Comparing the results of both training runs, one of the two approaches was chosen for further research. The second part was to find the optimal model architecture and configuration parameters for the classifier by optimizing the hyperparameters using the Hyperopt library.

The performance of the static model without class weights is very stable with a PR score of 0.97718. The classification reports show that the positive class has a much higher (0.992) and a lower recall (0.939). This effect can be caused mainly by numerous negative class monsters in the training set. Consequently, the confusion matrix in Figure 12 reveals a few false positives, but a high number of false negatives.

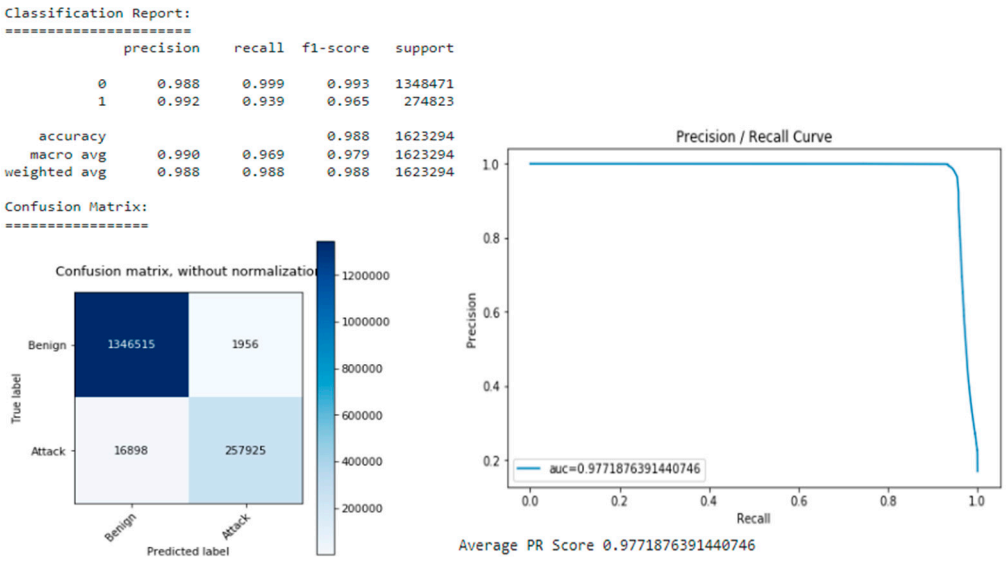


Figure 12. Static Model without class weights.

The misclassification statistics show that most of the false-negative results are due to the infiltration of the attack category, with a misclassification rate of 98%.

The performance of the static weighted class model for PR points is 0.97735 points, which is slightly better than the previous model (Figure 13).

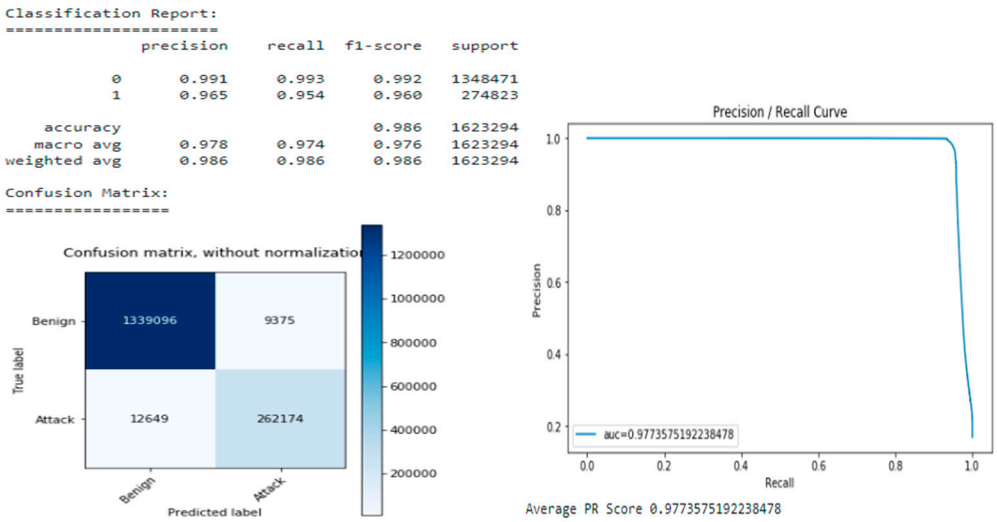


Figure 13. Static Model using class weights.

Nonetheless, the classification report shows a different combination of precision and recall for the positive class, with a precision of 0.965 and a recall of 0.954. The confusion matrix reveals an almost equal number of false positives (9375 samples) and false negatives (12,649 samples). Although the misclassification rate of infiltration was reduced to 77%, it still is very high. By using class weights during training, it is possible to reduce precision by about 2.7%, but increase recall by 1.6%. As a result, the number of false negatives decreases, and the number of false positives increases by an acceptable amount (Table 3).

Table 3. DNN models comparison.

Model	PR Score	Precision Positive	Recall Positive	False-Positives	False-Negatives
Static model (no class-weights)	0.97718	**0.992**	0.939	**1956**	16,898
Static model (class-weights)	0.97735	0.965	**0.954**	9375	12,649
Optimized model (class-weights)	**0.97816**	0.967	**0.954**	8966	**12,629**

Although the second approach does not look so appealing, the authors had to choose it because, at the beginning of the third experiment, the goal was to detect as many attacks as possible by using class weights during training. Additionally, both models suffer from a high level of misclassification against the penetration attack category, likely due to the fact that benign traffic and penetrating traffic are very similar.

In this section, model training with hyperparameter optimization was applied. The Hyperopt library [89] helped to train and evaluate different model architectures in combination with different hyperparameter configurations. For this purpose, it was necessary to decide on a training method that would accept model parameters from the Hyperopt library, dynamically create a model, perform training, and return minimal validation losses.

The best model has the following parameter configuration (Figure 14):

- 5 layers
- 300 units per layer
- A dropout rate of 0.22
- Elu activation function

- Adam optimizer
- A learning rate multiplier of 0.61 effectively reduces the default learning rate of 0.001 to 0.00061.

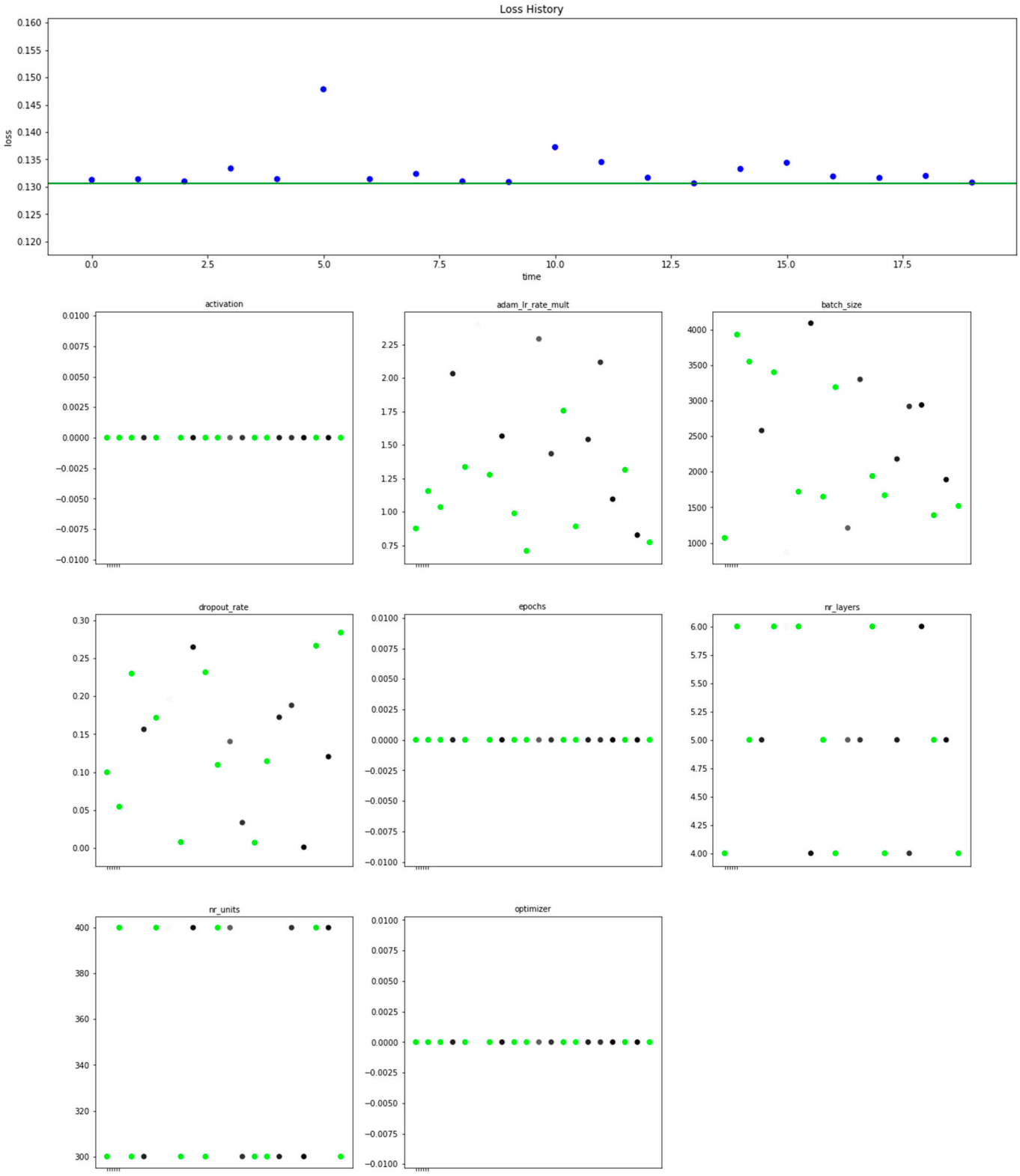


Figure 14. DNN Model Training with Hyperparameter Optimization.

The authors also wanted to perform a second optimization step using the optimal parameter values found in the first round of hyperparameter search. This would allow others to further explore the space of optimal parameters.

For this round, a variable batch size was used, and 20 trials were run. However, the second round of hyperparameter search did not give a better result. In the second optimization, the model loss was fixed at 0.1306, compared to the 0.1302 loss obtained from the best model of the first round. On the other hand, the difference in losses was negligible, which suggests that this is a fairly good configuration for the model.

In the last step of this section, the optimal parameter values and model configuration were used, which had been determined during the first round of hyperparameter searches. The model was trained using the optimal parameters for 200 epochs, with the objective of obtaining better performance. Inspecting the learning curves, it can be seen that the model does not significantly overfit the training set (Figure 15).

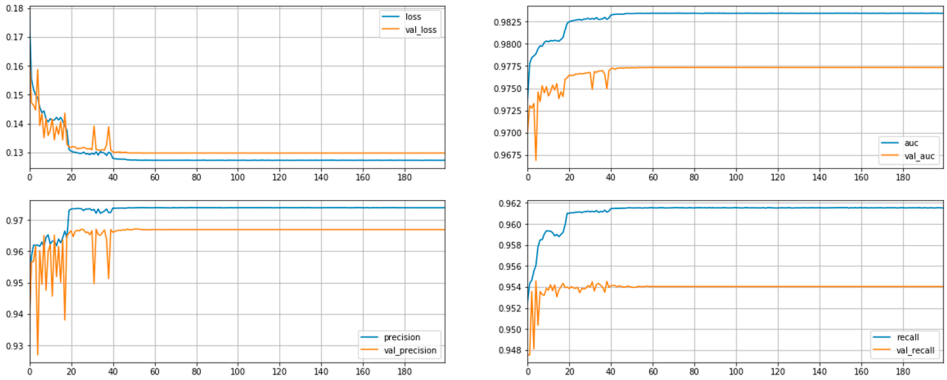


Figure 15. Learning curves of optimized model.

After training, a model was found with a slightly lower loss of 0.1298 and a better PR score of 0.97816 compared with the best model found when searching for hyperparameters with a loss of 0.1302 and a PR score of 0.97793 (see Figure 16).

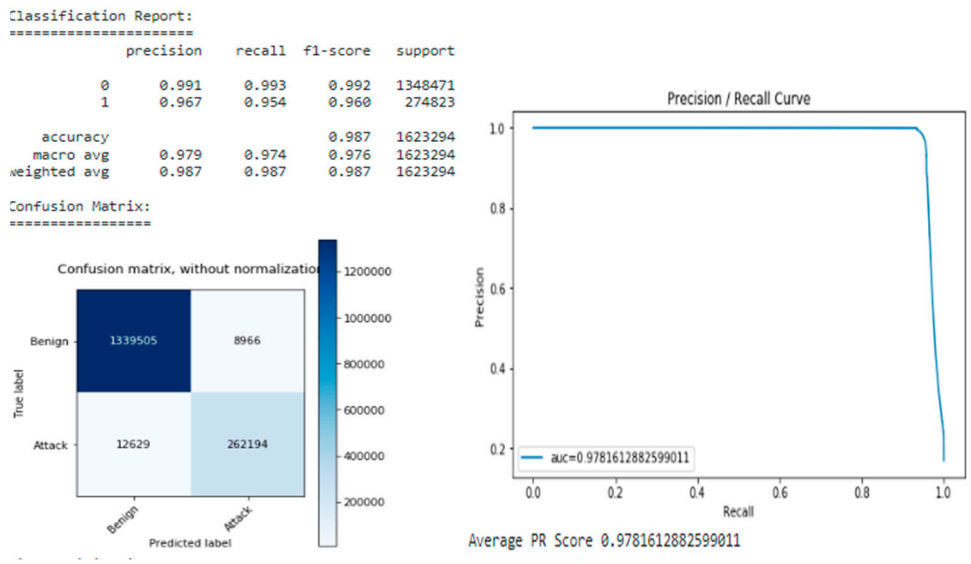


Figure 16. Model with Optimal Parameters.

To summarise this experiment, two approaches to train a deep neural network for the task of binary classification of network traffic were investigated. It is important that the

training was carried out whilst taking into account the class imbalance in order to minimize the number of false negative results.

In addition, we performed hyper parametric search and optimization to study the optimal network architecture and configuration of parameters, which resulted in a model with impressive performance.

Despite the fact that the model seems to work well with most types of network attacks, it cannot correctly identify penetration attacks, mistakenly classifying 77% of all penetration traffic as secure traffic. This can be explained by the similarity of the statistics of characteristics observed in these two types of network traffic.

As a result, it is necessary to explore other approaches for the reliable detection of penetration attacks.

3.7. Results

Ultimately, in order to compare which algorithm is the best to use for intrusion detection systems, a comparative analysis of the best three algorithms was carried out:

- Random forest classifier using scikit-learn
- Gradient Boosted Tree Classifier using CatBoost library
- Deep neural network using Keras and Tensorflow.

Initially, all models should be in the same positions, therefore the search and optimization of hyperparameters for the Random Forest and Gradient Boosted Tree algorithms were carried out, since in previous experiments the default parameters of both algorithms were used for training. As in previous experiments, the Hyperopt library was used to search by hyperparameters. The data used to train and compare models is from the CIC-IDS-2018 dataset.

After training and fine-tuning, the authors compared the models based on their respective characteristics in the validation set and selected the model with the best performance. This model was subsequently evaluated on a test set to obtain an unbiased estimate of the model's performance (Figure 17).

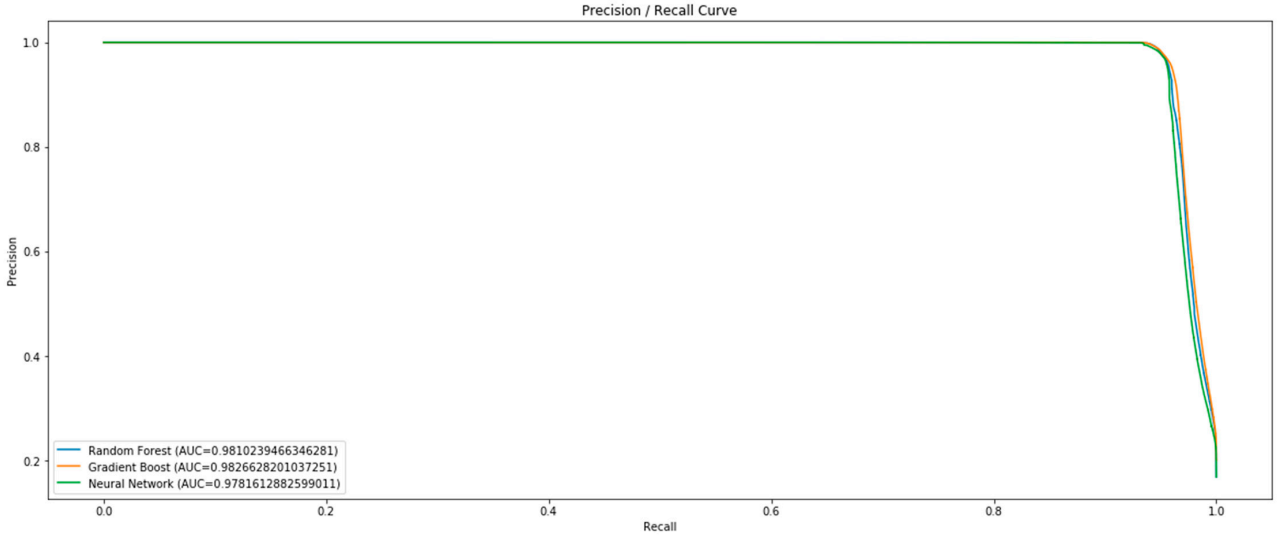


Figure 17. Precision/Recall curves of the different models.

The Random Forest and Gradient Boosted Tree models outperformed the neural network, achieving slightly higher PR, precision, and recall (Table 4 Comparison of models based on metrics).

Table 4. Comparison of models based on metrics.

Model	PR Score	Precision Positive	Recall Positive	False-Positives	False-Negatives
Random Forest	0.98102	**0.967**	0.955	**8820**	12,322
Gradient Boosted Trees	**0.98266**	0.964	**0.957**	9784	**11,748**
Deep Neural Network	0.97816	**0.967**	0.954	8966	12,629

Overall, the Gradient Boosted Tree model offered the best performance, with the highest PR and recall rate compared with the positive class, with only slightly lower accuracy than the random forest model. Therefore, this model also returned the fewest false negatives.

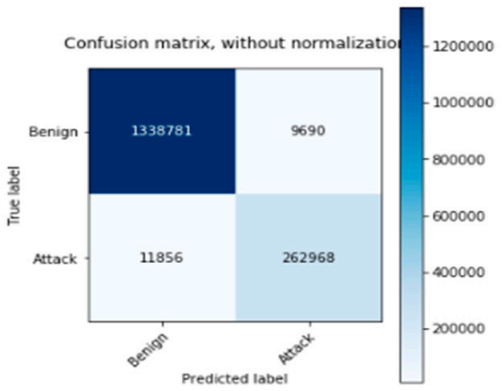
When the most efficient model algorithm was found, it had to be tested and evaluated objectively. As a result, if the performance on the test set was very similar to the performance on the first set, then the data has the same statistics as the test set and it can be assumed that this model generalizes unseen data well (see Figure 18 Performance on Test Set).

```
Classification Report:
======================
              precision    recall  f1-score   support

           0      0.991     0.993     0.992   1348471
           1      0.964     0.957     0.961    274824

    accuracy                          0.987   1623295
   macro avg      0.978     0.975     0.976   1623295
weighted avg      0.987     0.987     0.987   1623295

Confusion Matrix:
=================
```




```
Misclassifications by attack category:
======================================
```

	misclassified	total	percent_misclassified
Infilteration	11811	16194	0.729344
Brute Force -Web	3	61	0.049180
Brute Force -XSS	1	23	0.043478
DoS attacks-Slowloris	8	1099	0.007279
Benign	9690	1348471	0.007186
Bot	24	28619	0.000839
DDoS attacks-LOIC-HTTP	9	57619	0.000156

Figure 18. Performance on Test Set.

In our model's comparison experiment, the model was built using gradient boosting and tested on the test dataset and was found to perform well on the test data as well. The estimator shows a recall and accuracy of 0.98, which is acceptable for real use. In this study, minority attack classes are often misclassified. SMOTE can be applied to the training dataset for these classes to improve future performance. Additional tests can be performed on new data from different network environments to ensure that the estimator performs the same as it did on the test dataset.

4. Discussion and Research Challenges

With the rapid growth of the Internet of Things and the development of fifth generation mobile networks, there are great opportunities for further network cybersecurity research. The approach outlined in this research paper is just one of a few topics that address the security of the entire network. In addition, in order for an intrusion detection system to show good results, it is necessary to conduct further checks and tests on big data and from different gadgets/devices.

In general, during the study, it became clear that the machine learning approach allowed us to automate the process. This topic can be developed further, or an IDS can be created for the 5G core network using various machine learning or deep learning algorithms. It also generated ideas such as:

- The creation of a new dataset that will collect network traffic. This is very relevant, as some datasets have lost their novelty. After this set, one can research it and build new models;
- The implementation of a real-time traffic monitoring system;
- The application of a Machine Learning Approach to the Internet of Things;
- The protection of the machine learning system from potential hackers. For example, they can access 5G databases using vulnerabilities. Once they have access, they can use machine learning techniques to obtain sensitive information;
- The use of semi-supervised machine learning models—in reality, not many datasets have labeled data;
- The investigation of the possibility of detecting enemy attacks that lead to misleading predictions of unknown attack types;
- The observation of communication and security standards in mobile applications.

5. Conclusions

In this research paper, a comparative analysis of the application of machine learning and deep learning for a network intrusion detection system in a 5G network was carried out. However, prior to building the model, an overview was made of popular and recent datasets for tracking normal and malicious network traffic. As a result, two datasets were selected (CICIDS2017 and CSE-CIC-IDS2018).

The following models were trained and evaluated: Logistic Regression, Random Forest, Gradient Boosting, Autoencoder, and DNN with hyperparameter search. The whole process was divided into three parts and only one ML or DL algorithm was chosen in each experiment, meaning that in the end they could compete with each other. In the first part, the Gradient Boost algorithm performed well, showing a recall of 0.99 and a precision of 0.99. In the second part, when the autoencoder models were built, it was revealed that the performance of the resulting model was insufficient for real use and was much worse than the performance of the machine learning models. In the third part, two approaches for training a deep neural network for the task of binary classification of network traffic were explored.

In the end, according to the best results based on metrics, Gradient Boost came out as the best algorithm, showing on the test set 99.3% for a secure dataset and 96.4% for attacks.

An intrusion detection system (IDS) installed in a network examines network activity to learn about potential threats and vulnerabilities that could harm the system and the network environment. It is anticipated that IDS will be built into the core of the 5G network

as a new network feature. This methodology is also proposed for subsequent generations, 6G and 7G, as the architecture of this generation involves integrating advanced features into the existing 5G technology to perform tasks at the individual and group levels.

Author Contributions: Conceptualization, A.I., S.T. and R.O.; Data curation, R.B. and A.B.; Formal analysis, A.I., S.G. and N.K.; Investigation, A.I., N.K. and A.B.; Methodology, S.G., S.T. and A.I.; Project administration, R.B. and R.O.; Resources, A.I. and R.O.; Software, A.I. and N.K.; Supervision, S.T. and R.O.; Validation, S.G., A.B. and R.B.; Visualization, A.I. and R.O.; Writing—original draft, A.I., N.K. and R.O.; Writing—review and editing, S.T. and S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.unb.ca/cic/datasets/ids-2017.html, https://www.unb.ca/cic/datasets/ids-2018.html accessed on 14 December 2020.

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

Appendix A

Table A1. Network Intrusion Detection System datasets.

Dataset	General Information				Nature of the Data			Data Volume	Recording Environment	Evaluation	
	Year of traffic creation	Public Avail.	Normal Traffic	Attack Traffic	Metadata	Format	Anonymity	Count	Duration	Kind of traffic	Compl. Network
KDD CUP 99	1998	Yes	Yes	Yes	No	Other	None	5 M flows	Not specified	emulated	Yes
NSL-KDD	1998	Yes	Yes	Yes	No	Other	None	150 K flows	Not specified	emulated	Yes
UNSW-NB15	2015	Yes	Yes	Yes	Yes	Packet, other	None	2 M flows	31 hours	emulated	Yes
CICIDS2017	2017	Yes	Yes	Yes	Yes	Packet, bidirectional flow	None	5 M flows	10 days	emulated	Yes
CSE-CIC-IDS2018	2018	Yes	Yes	Yes	Yes	Packet, bidirectional flow	None	3.1 M flows	5 days	emulated	Yes

Appendix B

```
import numpy as np
import pandas as pd
import os
import re

csv_files = {
 'Wednesday-28-02-2018_TrafficForML_CICFlowMeter.csv': 'infiltration_28-02-2018.csv',
 'Thursday-01-03-2018_TrafficForML_CICFlowMeter.csv': 'infiltration_01-03-2018.csv',
 'Friday-02-03-2018_TrafficForML_CICFlowMeter.csv': 'bot_02-03-2018.csv',
 'Thursday-22-02-2018_TrafficForML_CICFlowMeter.csv': 'bruteforce-web-xss_sql-injection_22-02-2018.csv',
 'Thursday-15-02-2018_TrafficForML_CICFlowMeter.csv': 'dos-goldeneye-slowloris_15-02-2018.csv',
 'Thuesday-20-02-2018_TrafficForML_CICFlowMeter.csv': 'ddos-loic-http-loic-udp_20-02-2018.csv',
 'Wednesday-21-02-2018_TrafficForML_CICFlowMeter.csv': 'ddos-loic-udp_hoic_21-02-2018.csv',
 'Wednesday-14-02-2018_TrafficForML_CICFlowMeter.csv': 'bruteforce-ftp-ssh_14-02-2018.csv',
 'Friday-16-02-2018_TrafficForML_CICFlowMeter.csv': 'dos-slowhttp-hulk_16-02-2018.csv',
 'Friday-23-02-2018_TrafficForML_CICFlowMeter.csv': 'bruteforce-web-xss_sql-injection_23-02-2018.csv'
}

column_name_regex = re.compile(r"\W", re.IGNORECASE)
processed_dir = 'processed'
processed_path = os.path.join(dataset_base_path, processed_dir)

def remove_headers(f):
    return f[~f['Dst Port'].str.contains('Dst Port', na=False)]

def replace_infinity(f):
    return f.replace('Infinity', 'inf', inplace=True)

def remove_non_word_chars_from_column_names(f):
    return [column_name_regex.sub('_', c.lower()) for c in df.columns]

if not os.path.exists(processed_path):
    os.mkdir(processed_path)

for f, out in csv_files.items():
    file_path = os.path.join(dataset_base_path, f)
    output_path = os.path.join(dataset_base_path, processed_dir, out)

    df = pd.read_csv(file_path, dtype=str).drop(columns=['Flow ID', 'Src IP', 'Dst IP', 'Src Port'], errors='ignore')
    df = remove_headers(df)
    replace_infinity(df)
    df.columns = remove_non_word_chars_from_column_names(df)
    df.to_csv(output_path, index=False)
```

Figure A1. Script for data cleaning.

References

1. Woodland, L. The Importance of 5G Technology. Secure Communications. Airbus. Available online: https://securecommunications.airbus.com/en/meet-the-experts/the-importance-of-5g-technology (accessed on 21 April 2021).
2. Lohiya, R.; Thakkar, A. Application Domains, Evaluation Data Sets, and Research Challenges of IoT: A Systematic Review. *IEEE Internet Things J.* **2021**, *8*, 8774–8798. [CrossRef]
3. Saad, W.; Bennis, M.; Chen, M. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Netw.* **2019**, *34*, 134–142. [CrossRef]
4. Thakkar, A.; Lohiya, R. A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. *Arch. Comput. Methods Eng.* **2020**, *28*, 3211–3243. [CrossRef]
5. Casillas, R.; Touchette, B.; Tawalbeh, L.; Muheidat, F. 5G Technology Architecture: Network Implementation, Challenges and Visibility. *Int. J. Comput. Sci. Inf. Secur.* **2020**, *18*, 39–53.
6. Lohiya, R.; Thakkar, A. Intrusion detection using deep neural network with antirectifier layer. In *Applied Soft Computing and Communication Networks*; Springer: Singapore, 2021; Volume 187, pp. 89–105. ISBN 978-981-33-6172-0. [CrossRef]
7. Thakkar, A.; Lohiya, R. A review of the advancement in intrusion detection datasets. *Procedia Comput. Sci.* **2020**, *167*, 636–645. [CrossRef]
8. Lohiya, R.; Thakkar, A. Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system. *Int. J. Intell. Syst.* **2021**, *36*, 7340–7388. [CrossRef]
9. Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. AI and 6G Security: Opportunities and Challenges. In Proceedings of the 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Porto, Portugal, 8–11 June 2021; pp. 616–621. [CrossRef]
10. Abdellah, R.; Mahmood, O.A.K.; Paramonov, A.; Koucheryavy, A. IoT traffic prediction using multi-step ahead prediction with neural network. In Proceedings of the 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland, 28–30 October 2019; pp. 1–4. [CrossRef]
11. Veeramreddy, J.; Prasad, V.; Prasad, K. A Review of Anomaly based Intrusion Detection Systems. *Int. J. Comput. Appl.* **2011**, *28*, 26–35. [CrossRef]

12. Alrajeh, N.A.; Khan, S.; Shams, B. Intrusion Detection Systems in Wireless Sensor Networks: A Review. *Int. J. Distrib. Sens. Netw.* **2013**, *9*. [CrossRef]
13. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecur* **2019**, *20*, 2. [CrossRef]
14. Thakkar, A.; Lohiya, R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artif. Intell. Rev.* **2021**, *55*, 453–563. [CrossRef]
15. Li, J.; Zhao, Z.; Li, R. A Machine Learning Based Intrusion Detection System for Software Defined 5G Network. *arXiv* **2017**, arXiv:1708.04571. [CrossRef]
16. Chua, T.; Salam, I. Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection System. *arXiv* **2017**, arXiv:2203.05232. [CrossRef]
17. Rajput, D.; Thakkar, A. *A Survey on Different Network Intrusion Detection Systems and CounterMeasure. Emerging Research in Computing, Information, Communication and Applications*; Springer: Singapore, 2019; pp. 497–506. [CrossRef]
18. Imanbayev, A.; Tynymbayev, S.; Odarchenko, R.; Alikhankyzy, Z. An analysis of the security problems of fifth generation mobile networks. *Phys. Math. Sci. Ser. Phys. Math. Sci.* **2021**, *76*, 21–28. [CrossRef]
19. Afaq, A.; Haider, N.; Baig, M.Z.; Khan, K.S.; Imran, M.; Razzak, I. Khan, Muhammad Imran, Imran Razzak, Machine learning for 5G security: Architecture, recent advances, and challenges. *Ad. Hoc. Netw.* **2021**, *123*, 102667. [CrossRef]
20. ENISA. Threat Landscape for 5G Networks. Available online: https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-for-5g-networks/@@download/fullReport (accessed on 14 December 2020).
21. Lakhbir, K.; Chitender, K. Security Survey and Study of DDos Attack on LTE (4G) Network. *IJRECE* **2016**, *4*. [CrossRef]
22. Henrydoss, J.; Boult, T. Critical security review and study of DDoS attacks on LTE mobile network. In Proceedings of the 2014 IEEE Asia Pacific Conference on Wireless and Mobile, Bali, Indonesia, 28–30 August 2014; pp. 194–200. [CrossRef]
23. ENISA. NFV Security in 5G—Challenges and Best Practices. Available online: https://www.enisa.europa.eu/publications/nfv-security-in-5g-challenges-and-best-practices/@@download/fullReport (accessed on 24 February 2022).
24. You, X.; Zhang, C.; Tan, X.; Jin, S.; Wu, H. Ai for 5g: Research directions and paradigms. *Sci. China Inf. Sci.* **2019**, *62*, 21301. [CrossRef]
25. Pawlicki, M.; Choras, M.; Kozik, R. Defending network intrusion detection systems against adversarial evasion attacks. *Future Gener. Comput. Syst.* **2020**, *110*, 148–154. [CrossRef]
26. Haider, N.; Baig, M.Z.; Imran, M. Artificial Intelligence and Machine Learning in 5G Network Security: Opportunities, advantages, and future research trends. *arXiv* **2020**, arXiv:2007.04490v1. [CrossRef]
27. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2224–2287. [CrossRef]
28. Kaur, J.; Khan, M.A.; Iftikhar, M.; Imran, M.; Haq, Q.E.U. Machine Learning Techniques for 5G and Beyond. *IEEE Access* **2021**, *9*, 23472–23488. [CrossRef]
29. Thakkar, A.; Lohiya, R. Role of swarm and evolutionary algorithms for intrusion detection system: A survey. Swarm and Evolutionary Computation. *Swarm Evol. Comput.* **2019**, *53*, 100631. [CrossRef]
30. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [CrossRef]
31. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine learning for resource management in cellular and IoT networks: Potentials current solutions and open challenges. *IEEE Commun. Surveys Tuts* **2020**, *22*, 1251–1275. [CrossRef]
32. Yao, M.; Sohul, M.; Marojevic, V.; Reed, J.H. Artificial intelligence defined 5g radio access networks. *IEEE Commun. Mag.* **2019**, *57*, 14–20. [CrossRef]
33. Kumar, G.; Kumar, K.; Sachdeva, M. The use of artificial intelligence based techniques for intrusion detection: A review. *Artif. Intell. Rev.* **2010**, *34*, 369–387. [CrossRef]
34. Kumar, G.; Kumar, K. The Use of Artificial-Intelligence-Based Ensembles for Intrusion Detection: A Review. *Appl. Comput. Intell. Soft Comput.* **2012**, *2012*, 850160. [CrossRef]
35. Kim, A.; Park, M.; Lee, D.H. AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection. *IEEE Access* **2020**, *8*, 70245–70261. [CrossRef]
36. Zamani, M.; Movahedi, M. Machine learning techniques for intrusion detection. *arXiv* **2013**, arXiv:1312.2177. [CrossRef]
37. An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2339–2349. [CrossRef]
38. Aneja, M.J.S.; Bhatia, T.; Sharma, G.; Shrivastava, G. Artificial intelligence based intrusion detection system to detect flooding attack in VANETs. In *Handbook of Research on Network Forensics and Analysis Techniques*; IGI Global: Hershey, PA, USA, 2018; pp. 87–100.
39. Choi, I.; Lee, J.; Kwon, T.; Kim, K.; Choi, Y.; Song, J. An Easy-to-use Framework to Build and Operate AI-based Intrusion Detection for In-situ Monitoring. In Proceedings of the 2021 16th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Republic of Korea, 19–20 August 2021; pp. 1–8. [CrossRef]
40. Aminanto, E.; Kwangjo, K. Deep learning in intrusion detection system: An overview. In *Proceedings of the 2016 International Research Conference on Engineering and Technology (2016 IRCET)*; Higher Education Forum: Aspen, CL, USA, 2016.

41. Yang, X.S.; Sherratt, S.; Dey, N.; Joshi, A. (Eds.) Fourth International Congress on Information and Communication Technology. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; Volume 1041. [CrossRef]
42. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
43. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
44. Ring, M. A Survey of Network-Based Intrusion Detection Data Sets. arXiv.Org. 2019. Available online: https://arxiv.org/abs/1903.02460 (accessed on 6 March 2019).
45. Iavich, M.; Iashvili, G.; Gagnidze, A.; Khukhashvili, S.; Simonovi, S. Intrusion Detection System for 5G. *Sci. Pract. Cyber Secur. J* **2021**, *5*, 1–8.
46. Ullah, I.; Mahmoud, Q.H. A Two-Level Flow-Based Anomalous Activity Detection System for IoT Networks. *Electronics* **2020**, *9*, 530. [CrossRef]
47. Dhanabal, L.; Shantharajah, S.P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2015**, 446–452.
48. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. *arXiv* **2018**, arXiv:1806.03517.
49. Iashvili, G.; Iavich, M.; Bocu, R.; Odarchenko, R.; Gnatyuk, S. Intrusion Detection System for 5G with a Focus on DOS/DDOS Attacks. In Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 22–25 September 2021; pp. 861–864. [CrossRef]
50. View on 5G Architecture: 5G PPP Architecture Working Group. Available online: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-For-public-consultation.pdf (accessed on 20 March 2019).
51. Barakabitze, A.A.; Ahmad, A.; Mijumbi, R.; Hines, A. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Comput. Netw.* **2020**, *167*, 106984. [CrossRef]
52. Alotaibi, D.; Thayananthan, V.; Yazdani, J. The 5G network slicing using SDN based technology for managing network traffic. *Procedia Comput. Sci.* **2021**, *194*, 114–121. [CrossRef]
53. Syed-Yusof, S.K.; Numan, P.E.; Yusof, K.M.; Bin Din, J.; Bin Marsono, M.N.; Onumanyi, A.J. Software-Defined Networking (SDN) and 5G Network: The Role of Controller Placement for Scalable Control Plane. In Proceedings of the IEEE International RF and Microwave Conference (RFM), Seremban, Malaysia, 12–14 December 2020; pp. 1–6. [CrossRef]
54. Bocu, R.; Iavich, M.; Tabirca, S. A Real-Time Intrusion Detection System for Software Defined 5G Networks. In *Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2021; pp. 436–446. [CrossRef]
55. Ahmad, Z.; Khan, A.S.; Shiang, C.W.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2020**, *32*, e4150. [CrossRef]
56. Rawat, S.; Srinivasan, A.; Ravi, V.; Ghosh, U. Intrusion detection systems using classical ML techniques vs integrated unsupervised feature learning and deep neural network. *Internet Technol. Lett.* **2022**, *5*, e232. [CrossRef]
57. Lam, J.; Abbas, R. Machine Learning based Anomaly Detection for 5G Networks. *arXiv* **2020**, arXiv:2003.03474. [CrossRef]
58. Fredriksson, T.; Mattos, D.I.; Bosch, J.; Olsson, H.H. Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. In *Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2021; pp. 202–216. [CrossRef]
59. Cunningham, R.K.; Lippmann, R.P.; Fried, D.J.; Garfinkel, S.L.; Graf, I.; Kendall, K.R.; Webster, S.E.; Wyschogrod, D.; Zissman, M.A. *Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation*; Massachusetts Institute of Technology Lexington Lincoln Lab: Lexington, MA, USA, 1999.
60. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
61. UNSW Canberra at ADFA. The UNSW-NB15 Dataset. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 2 June 2021).
62. Hicham, M.; Abghour, N.; Ouzzif, M. 5G mobile networks based on SDN concepts. *Int. J. Eng. Technol.* **2018**, *7*, 2231. [CrossRef]
63. Maeder, A.; Ali, A.; Bedekar, A.; Cattoni, A.F.; Chandramouli, D.; Chandrashekar, S.; Du, L.; Hesse, M.; Sartori, C.; Turtinen, S. A Scalable and Flexible Radio Access Network Architecture for Fifth Generation Mobile Networks. *IEEE Commun. Mag.* **2016**, *54*, 16–23. [CrossRef]
64. Sarigiannidis, P.; Lagkas, T.; Bibi, S.; Ampatzoglou, A.; Bellavista, P. Hybrid 5G optical-wireless SDN-based networks, challenges and open issues. *IET Netw.* **2017**, *6*, 141–148. [CrossRef]
65. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information SystemsSecurity and Privacy, ICISSP 2018, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
66. Lin, P.; Ye, K.; Xu, C.-Z. Dynamic Network Anomaly Detection System by Using Deep Learning Techniques. In *What is Exploratory Data Analysis?* Towards Data Science: Toronto, ON, Canada, 2019; pp. 161–176. [CrossRef]

67. Abdulhammed, R.; Musafer, H.; Alessa, A.; Faezipour, M.; Abuzneid, A. Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. *ResearchGate* **2019**, *8*, 322. [CrossRef]
68. Ozkan-Okay, M.; Samet, R.; Aslan, O.; Gupta, D. A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access* **2021**, *9*, 157727–157760. [CrossRef]
69. Ridzuan, F.; Zainon, W.M.N.W. A Review on Data Cleansing Methods for Big Data. *Procedia Comput. Sci.* **2019**, *161*, 731–738. [CrossRef]
70. Patil, P. *What is Exploratory Data Analysis?* Towards Data Science: Toronto, ON, Canada, 2018.
71. Rao, A.S.; Vardhan, B.V.; Shaik, H. Role of Exploratory Data Analysis in Data Science. In Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES), Beijing, China, 17–19 June 2021; pp. 1457–1461. [CrossRef]
72. Tarwani, K.M.; Saudagar, S.S.; Misalkar, H.D. Machine learning in big data analytics: An overview. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2015**, *5*, 270–274.
73. Zaslavsky, A.; Perera, C.; Georgakopoulos, D. Sensing as a service and big data. In Proceedings of the International Conference on Advances in Cloud Computing (ACC), Bangalore, India, 26–28 July 2012; pp. 1–8.
74. Zafarani, R.; Abbasi, M.A.; Liu, H. *Social Media Mining: An Introduction*; Cambridge University Press: Cambridge, UK, 2014.
75. Shalev-Shwartz, S. Online learning and online convex optimization. *Found. Trends Mach Learn.* **2011**, *4*, 107–194. [CrossRef]
76. Wang, J.; Zhao, P.; Hoi, S.C.H.; Jin, R. Online feature selection and its applications. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 698–710. [CrossRef]
77. Bilenko, M.; Basu, S.; Sahami, M. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), Houston, TX, USA, 27–30 November 2005; p. 8.
78. Yu, C.H. Exploratory data analysis in the context of data mining and resampling. *Int. J. Psychol. Res.* **2010**, *3*, 9–22. [CrossRef]
79. Kumari, R.; Kr, S. Machine Learning: A Review on Binary Classification. *Int. J. Comput. Appl.* **2017**, *160*, 11–15. [CrossRef]
80. Naik, N.; Purohit, S. Comparative Study of Binary Classification Methods to Analyze a Massive Dataset on Virtual Machine. *Procedia Comput. Sci.* **2017**, *112*, 1863–1870. [CrossRef]
81. Peng, C.-Y.J.; Lee, K.L.; Ingersoll, G.M. An Introduction to Logistic Regression Analysis and Reporting. *J. Educ. Res.* **2002**, *96*, 3–14. [CrossRef]
82. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
83. Aziz, N.; Akhir, E.A.P.; Aziz, I.A.; Jaafar, J.; Hasan, M.H.; Abas, A.N.C. A Study on Gradient Boosting Algorithms for Development of AI Monitoring and Prediction Systems. In Proceedings of the 2020 International Conference on Computational Intelligence (ICCI 2022), Bandar Seri Iskandar, Malaysia, 8–9 October 2020; pp. 11–16. [CrossRef]
84. Pirie, W. Spearman Rank Correlation Coefficient. 2006. Available online: https://onlinelibrary.wiley.com/doi/10.1002/0471667196.ess2499.pub2 (accessed on 1 November 2022).
85. Tezcan, B. Why Using a Dummy Classifier is a Smart Move. 2021. Available online: https://www.courses-for-you.com/courses/why-using-a-dummy-classifier-is-a-smart-move-by (accessed on 1 November 2021).
86. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363 [cs.LG]. [CrossRef]
87. Filion, G.J. The signed Kolmogorov-Smirnov test: Why it should not be used. *GigaScience* **2015**, *4*, 9. [CrossRef] [PubMed]
88. Trevisan, V. *Using SHAP Values to Explain How Your Machine Learning Model Works*; Towards Data Science: Toronto, ON, Canada, 2018.
89. Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D.D. Hyperopt: A Python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **2015**, *8*, 14008. [CrossRef]