electronics

MDPI

Article

Unknown Security Attack Detection Using Shallow and Deep ANN Classifiers

Malek Al-Zewairi [1,*], **Sufyan Almajali** [1] **and Moussa Ayyash** [2]

[1] Department of Computer Science, Princess Sumaya University for Technology, Amman 11941, Jordan; s.almajali@psut.edu.jo

[2] Department of Computing, Information, and Mathematical Sciences and Technology, Chicago State University, Chicago, IL 60628, USA; msma@ieee.org

* Correspondence: m.alzewairi@jisdf.org

Received: 5 October 2020; Accepted: 22 November 2020; Published: 26 November 2020

Abstract: Advancements in machine learning and artificial intelligence have been widely utilised in the security domain, including but not limited to intrusion detection techniques. With the large training datasets of modern traffic, intelligent algorithms and powerful machine learning tools, security researchers have been able to greatly improve on the intrusion detection models and enhance their ability to detect malicious traffic more accurately. Nonetheless, the problem of detecting completely unknown security attacks is still an open area of research. The enormous number of newly developed attacks constitutes an eccentric challenge for all types of intrusion detection systems. Additionally, the lack of a standard definition of what constitutes an unknown security attack in the literature and the industry alike adds to the problem. In this paper, the researchers reviewed the studies on detecting unknown attacks over the past 10 years and found that they tended to use inconsistent definitions. This formulates the need for a standard consistent definition to have comparable results. The researchers proposed a new categorisation of two types of unknown attacks, namely Type-A, which represents a completely new category of unknown attacks, and Type-B, which represents unknown attacks within already known categories of attacks. The researchers conducted several experiments and evaluated modern intrusion detection systems based on shallow and deep artificial neural network models and their ability to detect Type-A and Type-B attacks using two well-known benchmark datasets for network intrusion detection. The research problem was studied as both a binary and multi-class classification problem. The results showed that the evaluated models had poor overall generalisation error measures, where the classification error rate in detecting several types of unknown attacks from 92 experiments was 50.09%, which highlights the need for new approaches and techniques to address this problem.

Keywords: unknown attacks; network anomaly; intrusion detection; IDS; deep learning

1. Introduction

Since the mid-1960s, the research community emphasised the importance of monitoring computer systems in order to detect intrusive and malicious activities. Currently, security attacks have become more sophisticated and difficult to detect due to the advancement in network technologies, the prevalence of encryption and the fact that applications are emerging at an extremely rapid pace. On average, more than 4100 new apps are released daily on the Apple App Store and Google Play Store combined [1], making the task of distinguishing between beginning and intrusive traffic in order to detect new security attacks, such as zero-day attacks, a challenging issue, especially for the traditional techniques of detecting intrusions that depend on pre-existing knowledge of the attack, such as signature-based methods, thus making them obsolete [2–4].

The intrusion detection system (IDS) plays an essential role in discovering potential security threats and violations. Its main function is to continuously monitor computer and network systems for signs of intrusive activities, and it is an integral part of the defence-in-depth security paradigm. Generally, intrusion detection systems are categorised based on their data source, detection technique, deployment architecture, deployment applications, anomaly type and defence mechanism [5–7]. A network anomaly detection system (NADS) is a type of anomaly-based IDS that applies machine learning (ML) and artificial intelligence (AI) techniques on network traffic in order to learn how to distinguish between anomalous and normal traffic [5].

The effectiveness of an IDS is measured by several metrics and is often dependent on whether a supervised or an unsupervised model is used, as well as whether a binary classification model or a multi-classification model is used [8]. Often, the false alarm rate (FAR) and the true alarm rate (TAR) are used to measure the performance of the IDS in detecting security attacks. The former refers to the percentage of benign events that are falsely identified as malicious, while the latter refers to the percentage of malicious events correctly flagged as threats. A perfect IDS has an FAR percentage of zero and a TAR of 100%; meaning that, the IDS identifies all anomalous and normal activities correctly without any misclassification. However, realistically, this goal is hard to achieve, if not impossible [7].

Similarly, the performance of an NADS is dependent on its ability to correctly classify unseen traffic from previously seen traffic, which is referred to as generalisation error. Ultimately, this is a factor of the quality of the dataset, the output of the features engineering stage and the selected anomaly detection algorithm [6].

Intrusion and anomaly detection systems face various challenges that researchers are ceaselessly trying to address. These open issues have been extensively identified in several surveys [6,7,9,10]. The lack of standard datasets that are reliable, realistic and publicly available is considered one of the major open issues for IDS/NADS. Having inaccurately labelled datasets with poorly diversified attack scenarios in addition to noisy, irrelevant and incomplete data can badly affect the credibility of the dataset. Additionally, the utilisation of different evaluation metrics makes the process of comparing different techniques impossible. Furthermore, designing an effective IDS/NADS has become a more complex task to address due to the increased complexity of network systems, as well as attack techniques. Moreover, the real-time detection of anomalous traffic is a major challenge due to the huge increase in traffic volume and speed of modern computer networks [11]. In addition to the aforementioned challenges, detecting unknown attacks has proven to be a challenging issue for several reasons as follows:

- Firstly, anomaly-based intrusion detection techniques are dependent on the conceptualisation of normality (i.e., what constitutes a normal or legitimate profile), which is a very challenging issue by itself.
- Secondly, defining an inclusive profile of normal traffic is usually not feasible due to the complex similarities between normal and anomalous traffic.
- Thirdly, the rapid change of anomalous behaviours either by introducing completely new attacks or enhancing already existing attacks requires a dynamically adaptable solution, which can greatly impact the performance of IDS/NADS and, at the same time, is not an easy task.
- Finally, the prevalent utilisation of encryption in network traffic and applications in addition to the rapid increase in ransomware attacks create further challenges for the current intrusion and anomaly detection techniques.

In the current era of emerging attacks and advanced persistent threats (APTs), the ability to detect unknown attacks that the IDS/NADS has never encountered before has become a necessary feature for any intrusion detection technique. Over the past 10 years, several works in the literature have tried to address the issue of detecting unknown attacks. Nonetheless, we found that many of the proposed techniques lack rigorous evaluation, either empirical or formal; most importantly, they present an inconsistent definition of what constitutes an unknown attack, where the majority of works incorrectly referred to unknown attacks as unseen instances in the traditional ML sense—that is, a seen class, but unseen instance. Evidently, this is a flawed way to approach the problem. The authors identify this as an additional open research issue that is worth investigating. Here, the authors aim to address the issue of detecting unknown attacks by proposing a standard categorisation for what constitutes an unknown attack. This includes both the unknown category and the subcategory of security attacks. Then, the proposed categorisation is used to test the ability of previously proposed NADS models that employ both deep and shallow artificial neural network (ANN) classifiers to detect unknown attacks under the proposed categorisation.

The rest of the paper is organised as follows: Section 2 highlights the state-of-the-art in the literature in terms of detecting unknown attacks. Section 3 presents the research methodology including the proposed categorisation for unknown attacks. Section 4 discusses the dataset selection process and the experimental analysis of the research problem. Section 5 presents the comparative analysis results. Finally, in Section 6, the paper is concluded, and future work is presented.

2. Literature Review

This section highlights the papers that aimed to address the issue of detecting modern unknown network intrusions in the past 10 years, as well as several recently proposed novel anomaly-based intrusion detection models.

Kukiełka and Kotulski (2010a) proposed an IDS classifier based on a multi-layer perceptron (MLP) model with the aim of detecting new types of attacks that were not defined in the knowledge discovery and data mining (KDD99) dataset. The model was trained to identify 22 types of attacks and was tested on a modified dataset, which included 14 new types of attacks that were manually generated using the Metasploit Framework [12]. Nonetheless, all the newly generated attacks were new variants of the buffer overflow attack, which the model was already trained to identify; therefore, they were not truly new attacks. In addition, the proposed model was not able to detect the attacks until it was trained on a new feature (i.e., the "longest string sent") that was specifically added to allow the model to identify the new variants. This implies that the model had to be trained manually to recognise such attacks.

Kukiełka and Kotulski (2010b, 2014) continued to improve on their previously proposed model in [12] by experimenting with the support vector machine (SVM) ML algorithm [13,14]. Similarly, the model had to be trained manually to detect new attacks by adding new features. The model performed poorly in cases in which the attack pattern highly resembled normal traffic (e.g., SNMP get and guess attacks), thus rendering the proposed model inapplicable.

Bao et al. (2015) highlighted the lack of a well-defined methodology to detect new attacks and proposed a general reasoning methodology to infer new attacks from their factors. The authors assumed that each attack consists of several basic factors that occur in a specific pattern and that some attacks share similar factors. Based on the aforementioned assumption, they proposed a forward deduction method that utilises strong relevant logic to infer new attacks. Finally, the authors defined six factors to identify new attacks, which are the attacker goals, methods and resources in addition to the target resources, authentication methods and countermeasures [15]. The proposed method was not analysed nor evaluated; moreover, the defined factors are not sufficient to detect modern attacks such as APTs. In addition, the authors did not provide an exact definition for what constitutes a new attack, nor a specific scope for which the proposed method can be used (network attacks, OS attacks, software vulnerabilities, etc.).

Bao et al. (2016) presented an evaluation for their previously proposed methodology in [15], focusing on the detection of unknown attacks on cryptographic protocols [16]. The authors utilised two basic mutual authentication protocols to demonstrate potential attacks on cryptography-based protocols in order to validate the proposed methodology. Nevertheless, the proposed method did not provide a rigorous evaluation other than the common manual testing of cryptographic functions.

Ajjouri et al. (2016) proposed a distributed hierarchical agent-based IDS architecture to learn new patterns of security attacks using the case-based reasoning technique [17]. The proposed technique relies on detecting new attacks by learning from the similarities with known attacks. The proposed architecture was neither evaluated empirically nor formally, which makes it unverifiable.

Sellami et al. (2017) proposed cloud, agent and anomaly-based IDS. The proposed model works by pushing an agent to the device trying to connect to the cloud service provider (CSP). This agent acts as an IDS or an intrusion prevention system (IPS). Once the user is successfully authenticated, the behaviour is compared to a predefined normal profile to detect potential anomalies by calculating the projection distance between the observed behaviour and the predefined normal behaviour. An anomaly event is triggered only when this distance is greater than a certain threshold [18]. The proposed model resembles the ordinary agent-anomaly-based IDS. In addition, the model's ability to detect new attacks is merely based on the definition of anomaly-based IDS and was presented without empirical evaluation.

Aljawarneh et al. (2017) proposed a hybrid anomaly-based IDS that uses seven classifiers, namely J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump and naive Bayes (NB), in a voting-based ensemble learning model to improve the detection accuracy of said classifiers [19]. The proposed model utilises mutual information-based feature selection to reduce the number of features selected to train the model. The authors approached the problem as both a binary classification and multi-class classification problem using the NSL-KDD dataset and compared their results with J48, SVM and NB. The proposed model showed improved results when comparing the three algorithms under both classification problems (i.e., binary and multi-class).

Meira (2018) presented an experimental study into the detection of unknown attacks using unsupervised learning techniques. The study relied on the hypothesis that unknown attacks will appear as outliers. Four algorithms were evaluated—autoencoder neural network, K-means, K-nearest neighbour (k-NN) and isolation forest—using two datasets (i.e., NSL-KDD and ISCX-IDS). The datasets were processed to separate the normal traffic for training and the abnormal traffic for testing. The results showed that all algorithms performed poorly in terms of precision. The recall value was higher when compared with the precision [20]. The proposed technique utilises a one-class classification model, which means that the classifier only learns the normal traffic, and any variation in testing is considered an anomaly. Several attacks can share similar characteristics with normal traffic; therefore, these will be misclassified, which is evident from the poor precision.

Amato et al. (2018) proposed the use of an MLP classifier on the KDD99 dataset and explored the use of KNIME software for features selection and data preprocessing [21]. Although the authors stated that the proposed model is capable of identifying new types of security attacks without retraining the entire network, their work did not provide any experimental or theoretical analysis to support this.

Ahmad et al. (2018) addressed the efficiency issue of using machine learning algorithms, namely SVM, random forest (RF) and extreme learning machine (ELM), to handle large datasets in training intrusion detection classifiers [22]. The authors evaluated the algorithms on the NSL-KDD dataset using two splits: 90% and 80% for the training set, with 10% and 20% for the testing set, respectively. Their results suggested that ELM has better performance when presented with large data, while SVM performed the best with relatively small data.

Santikellur et al. (2019) proposed and evaluated the use of a new multi-layer network-based IDS to improve the detection rate of modern network attacks. The proposed system implements a hierarchical architecture of multiple machine learning models optimised using evolutionary computing algorithms organised into a two-layer architecture. The first layer consists of several variants of three binomial classifiers (i.e., AdaBoost, ANN and NB), while the second layer has a decision tree (DT) multi-class classifier [23]. The authors evaluated the system using the CIC-IDS-2017 dataset. Although the authors stated that the proposed approach can significantly improve the detection rate of modern unknown attacks, they did not evaluate the system performance when faced with an unknown attack.

Qureshi et al. (2019a) studied the performance of using a random neural network (RNN) classifier to detect anomalous traffic in the IoT environment [24]. The authors compared the proposed model with seven other machine learning algorithms using the NSL-KDD dataset and evaluated its performance using the full features and selected features based on the feature-selection techniques highlighted in [25]. The results showed that the proposed model greatly outperformed the other models in terms of precision and accuracy.

Qureshi et al. (2019b) proposed a hybrid anomaly-based intrusion detection model using RNN with the artificial bee colony (ABC) algorithm to find the optimal weights for the neurons [26]. The authors empirically evaluated the proposed model using the NSL-KDD dataset and under multiple learning rates. The results reported better performance measures for the proposed model when compared with traditional RNN models utilising the gradient descent algorithm.

Khare et al. (2020) proposed a hybrid model that combines the spider monkey optimisation (SMO) algorithm to reduce the dataset dimensionality with a deep neural network (DNN) classifier [27]. The proposed model showed a 3.3% improvement in the F1-score over a standalone DNN model with significantly reduced training time using the KDD99 dataset.

Khraisat et al. (2020) proposed a hybrid two-layer model that combines signature-based IDS with anomaly-based IDS with the aim of detecting known and unknown attacks [28]. The model uses the signature-based IDS to detect known attacks and then feeds the unknown signatures to the anomaly-based IDS to detect the unknown attacks.

Tang et al. (2020) proposed a deep learning IDS for SDN using binary DNN and gated recurrent neural network (GRU-RNN) classifiers [29]. The proposed model demonstrated better results in terms of the F1-measure when compared with the NB, DT and SVM classifiers using the NSL-KDD dataset.

Kim et al. (2020) designed a convolutional neural network (CNN) model to detect denial of service (DoS) attacks [30]. The authors utilised the KDD99 and the CSE-CIC-IDS-2018 datasets to train and test the model, where the comma separated value-based datasets were transformed into multidimensional vectors and then encoded into RGB and greyscale images to create new image-based datasets. The proposed design was applied as both a binary and multi-class classifier and then compared against an RNN-based classifier.

Jo et al. (2020) devised three field-to-pixel-based preprocessing techniques to transfer packet-based datasets into image-based datasets for IDS evaluation [31]. The proposed methods were used to transfer the NSL-KDD dataset into greyscale images and tested using a CNN-based model against known and unknown attacks.

In conclusion, it is apparent that, although many researchers have tried to address the issue of detecting unknown attacks [12–18,20,21,23,28,29,31], the proposed techniques either lacked rigorous analysis and evaluation [15–17,21] or had a flawed definition of unknown attacks [12–14,18,23]. The lack of a complete solution for the problem demonstrates the need for further research in this area. Table 1 summarises the related work and highlights its limitations.

3. Research Methodology

In this section, the research problem and methodology are identified and stated. Additionally, the new categorisation of unknown attacks is proposed, justified and explained.

In this study, the researchers started by surveying the literature from the past 10 years for studies that focused on detecting unknown security attacks. Thereafter, they found a need for the proposal of a standard categorisation of unknown attacks. Then, several common datasets that have been extensively used by the research community were studied and analysed with the goal of identifying which of them could be used to study unknown attacks under the proposed categorisation. Furthermore, two datasets and two ANN models were selected, and a series of experiments were conducted to study whether modern anomaly-based intrusion detection models are capable of detecting unknown attacks under the proposed categorisation. The research problem was addressed as both a binary classification and multi-class classification problem, which ensures that the results are more comprehensive.

Table 1. Summary of the related work.

Ref.	Proposed Solution	Testing Attacks Type	Datasets	Validation Techniques	Limitations
[12]	MLP classifier	Buffer overflow	Training: KDD99. Testing: 14 manually generated attacks.	ML testing	The model had to be retrained on new features. Not truly new attacks.
[14]	MLP and SVM classifier	Buffer overflow	Training: KDD99. Testing: 14 manually generated attacks.	ML testing	The model had to be retrained on new features. Poor performance when attacks resemble normal traffic.
[15]	Forward deduction method to infer new attacks	-	-	-	Lacks rigours evaluation
[16]	Forward deduction method to infer new attacks on cryptographic protocols	Masquerade. Bypass timestamp in session key matching.	-	Logical reasoning	Lacks rigours evaluation
[17]	Case-based reasoning technique	-	-	-	Lacks rigours evaluation
[18]	Cloud agent-based NADS	DoS	-	Manual simulation	Not truly new attacks
[20]	One-class classification model: Autoencoder ANN K-means Nearest neighbour Isolation forest	Several	NSL-KDD ISCX-IDS	ML testing	Poor precision (high FAR)
[21]	MLP classifier	DoS Privilege escalation Probing	KDD99	ML testing	Not truly new attacks
[23]	Multi-layer IDS: AdaBoost ANN Naive Bayes Decision tree	Several	CIC-IDS-2017	ML testing	Not truly new attacks
[28]	Hybrid two-layer IDS: C5 decision tree and SVM	Several	NSL-KDD ADFA	ML Testing	The results did not clearly highlight the model performance in detecting unknown attacks.
[29]	DNN and GRU-RNN classifiers	Several/SDN	NSL-KDD	ML testing	The dataset used is not built for SDN.

3.1. Problem Statement

Security attacks are becoming more targeted and less general, with an epidemic escalation in attack complexity [32]. Attackers are currently relying more on unknown attack techniques that resemble normal traffic and are fragmented, encrypted or obfuscated to avoid detection by traditional intrusion detection systems [33]. This is evident from the study of the statistics of the common techniques used in cyber-attacks published in monthly reports [34]. Figure 1 shows the distribution of the attack techniques from 2014 to Q2 2020. When comparing the top attack techniques over the past six years, two conclusions can be drawn: firstly, unknown techniques were the dominant attack vector for the years 2014–2016 and displayed a steady increase in the that period; secondly, unknown techniques witnessed a noticeable decrease in the period from 2017 to 2019. One might argue that this decrease is merely a result of attackers relying on easier attack vectors such as account hijacking, in addition to the rise of cyber-crime as a motivation behind security attacks from 62.3% in 2014 to 85.82% ending Q2 in 2020, which explains the prominent inflation in the number of malware-related attacks in the same period (i.e., 2017–2019). In addition, unknown attacks were the third most common kinds of security attacks in the Middle East and North Africa (MENA) region in March 2019 [35].

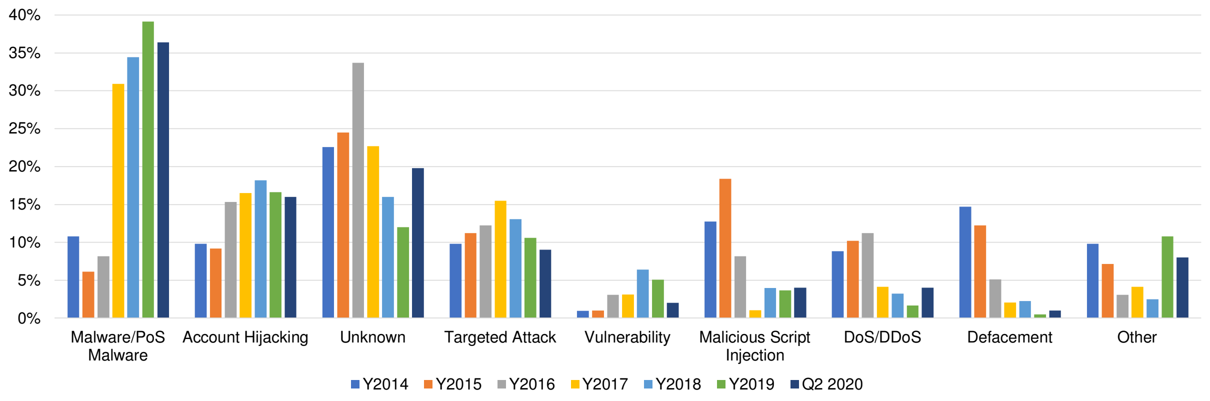

Figure 1. Attack distribution techniques from 2014 to Q2 2020.

When considering the different IDS types based on detection techniques and their ability to handle unknown attacks, signature-based IDS is inherently incapable of detecting either completely new attacks or any new variation of existing attacks. This is due to its design, where an attack is identified by matching it to a well-known signature. On the other hand, anomaly-based IDS is designed to learn normal behaviour and flags any variation as anomalous behaviour. This design makes the NADS more suitable for detecting unknown attacks. Nonetheless, it still suffers from several issues: fundamentally, it is not straightforward to define normal traffic, and there is a great intersection between the boundaries of normal and anomalous behaviour, making it a challenging task for NADS to detect unknown attacks. Conversely, specification-based IDS manually defines the characteristics of a specific network protocol and identifies any variation as an anomaly; thus, it is able to detect both known and unknown attacks. This technique is susceptible to enormous errors since not all applications strictly adhere to protocol specifications.

From the literature review, the researchers found that there is an apparent lack of a constant definition of what constitutes an unknown attack. We aimed to address this issue by proposing a standard categorisation to define an unknown attack and then studying the ability of modern ML models to detect unknown attacks under the proposed categorisation. In this study, the researchers focused on shallow and deep supervised ANN models in NADS.

3.2. Unknown Attacks

In this paper, the researchers propose a new categorisation of unknown attacks, comprising the unknown attack category and unknown attack subcategory:

- Unknown attack category: In terms of supervised machine learning, an unknown attack category refers to a completely new class label that the classifier has not encountered before in the training set; for example, if the classifier was not trained on DoS attacks and encountered DoS attacks in the testing set. The authors refer to this type of unknown attacks as Type-A.
- Unknown attack subcategory: This refers to a semi-new class label (i.e., a subclass); for example, if the classifier was indeed trained on detecting some DoS attack types, such as HTTP flooding, but encountered session initiation protocol (SIP) DoS attacks in the testing, which were not available in the training set yet share some characteristics with other DoS attacks present in the training set. The authors refer to this type of unknown attacks as Type-B.

The rationale behind the two-type proposed categorisation for unknown attacks detection is that anomaly-based classification requires the model to be trained on one or more datasets; however, having an all-inclusive dataset that contains all types of known attacks is not realistic. Moreover, unknown attacks are constantly ranked among the top attack distribution techniques since 2014 as shown in Figure 1. Furthermore, the proposed categorisation is consistent with how attacks are generally categorised in the real-world. For example: DoS attacks can occur from different types of intrusive activities, namely volumetric, resource exhaustion and application layer attacks. Meanwhile, each type of said attack constitutes multiple sub-types of the same attack. Internet control message protocol (ICMP), transmission control protocol (TCP), user datagram protocol (UDP) and internet protocol security (IPSec) flood attacks are examples of sub-types of DoS volumetric attacks. Additionally, having a consistent categorisation of unknown attacks helps to alleviate the confusion currently present in the literature; several researchers have incorrectly referred to new instances in a testing dataset or previously seen anomalies as unknown attacks. This categorisation provides a standard metric for evaluating works on unknown attack detection, which is currently lacking, as stated in [10].

4. Comparative Analysis

This section presents the comparative analysis to empirically study the ability of a modern NADS that utilises shallow and deep supervised ANN models to detect Type-A and Type-B unknown attacks under the proposed categorisation. Ninety-two experiments were conducted on two datasets (i.e., UNSW-NB15 and Bot-IoT) using deep and shallow ANN classifiers. With each experiment, both binary and multi-class supervised classifiers were built to identify the ability of the model to detect Type-A and Type-B of each attack class in the UNSW-NB15 and Bot-IoT datasets. Then, the accuracy, F1-score and classification error rate of unknown attack detection were calculated for each attack class for Type-A and for each subclass for Type-B. The results are presented and discussed in the Section 5.

4.1. Dataset Selection

The first stage of building an anomaly-based IDS includes the selection of a suitable dataset, which is used to train and evaluate the performance of the detection model [10]. The dataset has to be collected from a real network and correctly represent both benign and malicious network traffic to be representative. In nature, a true representative dataset is not balanced, because in reality, malicious traffic constitutes a small percentage of the overall traffic, which creates a challenge for the design of an accurate ML model [36].

Several datasets are available for evaluating IDS. Each dataset has its unique characteristics that make it suitable for certain types of research problems while unsuitable for others [36]. In order to find the correct datasets for this study, the researchers surveyed the most common and widely used datasets to indicate whether each dataset has both attack class and subclass labels, thus making it suitable for the study of unknown attacks. The results are shown in Table 2.

Table 2. Comparison between the most common datasets for IDS.

Dataset	**Year**	**Modern**	**Multi-Label**	**Label**				**Ratio**	
				Labels	**Overall**	**Class**	**Subclass**	**Attack**	**Normal**
KDD99 [37]	1999	N	N	Label	4,898,431	Normal DoS Remote-to-local (R2L) User-to-root (U2R) Probing	DoS: Back, Neptune, Pod, Smurf, Teardrop R2L: FTP Write, Guess Passwd, IMAP, Multihop, Phf, Spy, Warezclient, Warezmaster U2R: Buffer Overflow, Load-module, Perl, Rootkit Probing: IP-Sweep, Land, Nmap, Port-Sweep, Satan	75.00%	25.00%
NSL-KDD [38]	2009	N	Y	Attack Type Class	125,973	Attack Type: DoS R2L U2R Probing Class: Anomaly Normal	DoS: Back, Neptune, Pod, Smurf, Teardrop R2L: FTP Write, Guess Passwd, IMAP, Multihop, Phf, Spy, Warezclient, Warezmaster U2R: Buffer Overflow, Load-module, Perl, Rootkit Probing: IP-Sweep, Land, Nmap, Port-Sweep, Satan	46.54%	53.46%
ASNM-CDX [39]	2009	N	Y	Buffer Overflow Network Service	5771	Buffer Overflow: Attack (1) Benign (0) Network Service: Apache Postfix Other	Attack: Apache, Postfix Benign: Apache, Postfix, Other	00.76%	99.24%

Table 2. *Cont.*

Dataset	Year	Modern	Multi-Label	Label				Ratio	
				Labels	Overall	Class	Subclass	Attack	Normal
ISCX-IDS [40]	2012	Y	N	Tag	2,071,657	Attack Normal	Attack: DDoS IRC Botnet, DoS HTTP, Infiltration, SSH Brute Force	03.33%	96.67%
UNSW-NB15 [41]	2015	Y	Y	Category Label	2,540,046	Category: Analysis Backdoors DoS Exploits Fuzzers Generic Reconnaissance Shellcode Worms Label: Attack (1) Benign (0)	-	12.65%	87.35%
CIC-IDS-2017 [42]	2017	Y	N	Label	2,830,743	Benign Botnet Brute Force DDoS DoS Heartbleed Infiltration Port Scan Web	Brute Force: FTP, SSH DoS: GoldenEye, Hulk, SlowHTTPTest, Slowloris Web: Web Brute Force, SQL Injection, XSS	19.70%	80.30%

Table 2. *Cont.*

Dataset	Year	Modern	Multi-Label	Label				Ratio	
				Labels	Overall	Class	Subclass	Attack	Normal
NGIDS-DS [43]	2017	Y	Y	Category Subcategory Label	90,054,239	Category: Backdoors DoS Exploits Generic Recon Shellcode Worms Label: Abnormal Normal	Backdoors: All Batch DoS: Browser Batch, DCERPC, FTP, HTTP, Hypervisor, IIS, LDAP, Office Batch, Miscellaneous Batch, NetBIOS/SMB Batch, SMTP, SNMP, TCP, TFTP, Web Application XSS Batch, Windows Explorer Exploits: All, Backup Appliance, Browser, Browser FTP Batch, Client-side, Client-side Office Batch, Client-side Microsoft Paint, DCERPC Batch, DNS, ICMP, LDAP, IIS, IIS Batch, Miscellaneous Batch, MSSQL, NNTP, Office Document Batch, POP3, PPTP, RDesktop, SMB Batch, SMTP Batch, TFTP, Webserver, WINS Generic: IXIA Batch Reconnaissance: FrontPage HTTP Batch, IIS HTTP Shellcode: Linux Batch, Multiple OS Batch Worms: All Batch	01.40%	98.60%

Table 2. *Cont.*

Dataset	Year	Modern	Multi-Label	Label				Ratio	
				Labels	Overall	Class	Subclass	Attack	Normal
CSE-CIC-IDS-2018 [42]	2018	Y	N	Label	16,232,943	Benign Botnet Brute Force DDoS DoS Infiltration Web	Brute Force: FTP, SSH DDoS: HOIC, LOIC-HTTP, LOIC-UDP DoS: GoldenEye, Hulk, SlowHTTPTest, Slowloris Web: Web Brute Force, SQL Injection, XSS	16.93%	83.07%
Bot-IoT [44]	2018	Y	Y	Attack Category Subcategory	73,370,443	Attack: Attack (1) Normal (0) Category: DDoS DoS Theft Reconnaissance	DDoS: HTTP, TCP, UDP DoS: HTTP, TCP, UDP Theft: Data Exfiltration, Keylogging Reconnaissance: OS Fingerprint, Service Scan	99.99%	00.01%
IoT-23 [45]	2019	Y	Y	Label Detailed Label	325,309,946	Label: Benign Malicious	Detailed Label: Attack, C&C, C&C-FileDownload, C&C-HeartBeat, C&C-HeartBeat-Attack, C&C-HeartBeat-FileDownload, C&C-Mirai, C&C-PartOfAHorizontalPortScan, C&C-Torii, DDoS, FileDownload, Okiru, PartOfAHorizontalPortScan	90.51%	09.49%

In this study, the authors utilised two datasets: the UNSW-NB15 and the Bot-IoT datasets. Both datasets were generated by the same research laboratory using the same platform (i.e., Ixia PerfectStorm traffic generator), thus ensuring comparable results even though only Type-A unknown attacks can be generated from the UNSW-NB15 dataset. Moreover, they contain modern realistic network traffic with many benign and malicious attack types, as well as being validated and extensively studied by the research community. Furthermore, having both a network dataset and an IoT dataset ensures that the results can be generalised on both types of networks.

As described in Table 2, UNSW-NB15 has two labels, Category and Label, where the former is a multi-class label that hosts 10 classes, namely Analysis, Backdoors, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, Worms and Benign, whilst the latter is a binary class label with the value "0" indicating a benign instance and "1" an attack. The benign traffic constitutes 87% of the entire dataset, leaving 13% of the dataset as attack traffic distributed amongst the nine attack categories. Clearly, the dataset is not balanced; nonetheless, the real network traffic is not balanced either, as benign traffic is the predominant type of traffic [36].

On the other hand, the Bot-IoT has three labels; Attack, Category and Subcategory. The first is a binary class label with the value "0" indicating normal traffic and "1" attack traffic. The second is a multi-class label with four attack classes; DDoS, DoS, Reconnaissance and Theft, in addition to Normal. The third label is also a multi-class label, which contains seven attack classes: HTTP, TCP, UDP, Data Exfiltration, Keylogging, OS Fingerprint and Service Scan, in addition to the Normal class. In contrast with UNSW-NB15, the benign traffic in the Bot-IoT dataset constitutes less than 1% (i.e., 0.01%) of the entire traffic; therefore, subsampling had to take place. We used stratified sampling with a maximum cut of 9543 samples in order to include all the benign traffic and ensure a uniform distribution of the other attack labels. Table 3 shows the number of instances in the Bot-IoT before and after applying subsampling. It is worth mentioning that the Bot-IoT dataset has a smaller set, with 5% of the full dataset but with more features. In this work, the full dataset was used.

Table 3. Comparison between the number of instances of each class in the Bot-IoT dataset before and after applying stratified sampling with a maximum cut of 9543 samples per class.

Class	**Original**		**Sampling**	
	Count	**%**	**Count**	**%**
Normal	9543	0.01%	9543	10.91%
Data Exfiltration	118	0.00%	118	0.13%
Keylogging	1469	0.00%	1469	1.68%
OS Fingerprint	358,275	0.49%	9543	10.91%
Service Scan	1,463,364	1.99%	9543	10.91%
DoS HTTP	29,706	0.04%	9543	10.91%
DoS TCP	12,315,997	16.79%	9543	10.91%
DoS UDP	20,659,491	28.16%	9543	10.91%
DDoS HTTP	19,771	0.03%	9543	10.91%
DDoS TCP	19,547,603	26.64%	9543	10.91%
DDoS UDP	18,965,106	25.85%	9543	10.91%

4.2. Feature Engineering

Feature engineering involves the creation of a set of features from the collected network traffic to be used by the ML model to classify the traffic. This stage includes four major tasks: feature generation, feature selection, feature conversion and feature normalisation [10]. In feature generation, a collection of features is created from the raw captured network traffic. In feature selection, a subset from the generated features is selected to train the model. Association rules' mining, principal component analysis (PCA) and independent component analysis (ICA) are some of the techniques used in this stage [5]. Both tasks can greatly impact the overall performance of the ML model; general, poorly generated or selected features will result in a poorly performing model. In feature conversion, the format of the data is converted to be suitable for the selected algorithm (e.g., converting textual features into numerical features). Some ML libraries often perform this stage implicitly when needed. Finally, in feature normalisation, a function is used to scale the numerical values into a specific interval (e.g., [0–1]).

After the two datasets (i.e., UNSW-NB15 and Bot-IoT) were selected, they underwent several data preprocessing steps. The authors applied the same data preprocessing steps used in [46] to cleanse both datasets. Both datasets are published as multiple CSV files; therefore, the first step was to merge each dataset file into a single CSV file, then remove duplicate instances. The second step was to transform textual features into numerical values. In UNSW-NB15, three features were encoded (i.e., protocol, state and service), and in Bot-IoT, three features were also encoded (i.e., protocol, state and flags). The third step was to remove statistically insignificant features. In UNSW-NB15, the source and destination IP address features were removed, while in the Bot-IoT, 10 features where removed, namely packet sequence ID, timestamp, source/destination IP address, source/destination MAC address, source/destination country code, and source/destination organisationally unique identifier (OUI). In the fourth step, missing values were replaced with zero, then the numerical features where normalised into the interval [0, 1] using max-min normalisation. Finally, dataset instances were randomly shuffled.

4.3. Experimental Analysis

In this study, ninety-two experiments were conducted to test the ability of modern shallow and deep ANN models to detect unknown attacks under the proposed categorisation. Mainly, the experiments were divided into two parts: the first part involved testing the ability of shallow and deep multi-layer feed forward ANN models to detect unknown attacks of Type-A and Type-B under binary classification, whilst the second part involved the same test, but under multi-class classification.

Approaching the problem as both a binary and multi-class supervised classification problem ensured that the results were more representative. When testing for Type-A attacks, both datasets were used (i.e., UNSW-NB15 and Bot-IoT); however, only Bot-IoT was used to test Type-B attacks. In the literature, an ANN model with one/two hidden layers is referred to as a shallow model, whereas a model with two or more hidden layers is considered a deep structural model [7,47]. The general configurations for the shallow and deep ANN classifiers were proposed in [48] and [46], respectively, as shown in Table 4.

Table 4. Configuration parameters for shallow and deep ANN classifiers.

Parameters	**Shallow**	**Deep**
Hidden layers	**1**	**5**
Neurons per hidden layer	10	
Activation function	Tanh	
Batch size	32	
Dropout	Without dropout	
Epochs	10	
Fold assignment	Stratified assignment	
K-fold	10	
Learning rate	0.01	
Prediction threshold	0.5	
Results	Overall average of all runs	
Runs	30	
Seed	1,586,512,076,128	

Over the years, several ML algorithms have been utilised to build classification models for IDS/NADS, including but not limited to ANN, DT, K-means, k-NN, NB, RF and SVM. The use of these algorithms has been extensively studied in the literature [8,10,49,50], with each algorithm having its strengths, weaknesses and special requirements, thus making it suitable for certain types of applications. ANN models have been gaining remarkable recognition in the field of intrusion detection due to their uncanny ability to handle high-dimensional data and big data datasets [46,48–50]. It has been shown that ANN models outperform other ML models such as SVM, NB and RF [50]. Notably, the computational complexity and model performance increases with the number of ANN layers; therefore, deep ANN models have been shown to outperform shallow ANN models. In this study, the researchers opted to examine the performance of both deep and shallow ANN models at detecting unknown attacks in order to present representative results.

In this study, generating the testing sets is considered to be the most important step because it represents the basis for the evaluation of the proposed categorisation of unknown attacks; therefore, the first step in each experiment was to generate the training, validation and testing sets from each dataset. The testing sets were generated based on the class for Type-A attacks and subclasses for Type-B attacks in each dataset.

The procedure for generating the testing, training and validation sets for the binary classification model starts by defining a set of distinct attack classes for Type-A or subclasses for Type-B, which were extracted from the original dataset (e.g., DoS, DDoS, Reconnaissance and Theft for Type-A in the Bot-IoT dataset and Data Exfiltration, Keylogging, OS Fingerprint, Service Scan, HTTP, TCP and UDP for Type-B). Thereafter, all the instances of each class or subclass (based on the unknown attack type being generated) in the set were recursively extracted and saved as the testing set for that type of attack; meaning that the testing set only contains one class, which represents the unknown attack. Then, the remaining instances were split into training and validation sets using a fixed seed to ensure uniformity across all splits, with ratios of 0.9 and 0.1, respectively. This ensured that the testing set contained only one class of unknown attack and each corresponding training and validation set contained all the other benign and attack classes (except for that unknown attack).

For example, the testing set for the DoS attack of Type-A contained all the instances with a class label equal to DoS and nothing else, whilst the training and validation sets contained all the other instances in which the class label was not DoS. Following the same logic, the testing set for the UDP DoS attack of Type-B contained all the instances in which their class label was DoS and the subclass label was UDP and nothing else, whereas the training and validation sets contained all the other instances in which the class and subclass label were not DoS and UDP, respectively. Since the generated sets were used to evaluate binary classification models, both the class and subclass labels were removed, leaving only the binary class label. Conversely, when generating sets for evaluating multi-class classification models, the binary class label was removed, and the class label was kept for Type-A and the subclass label for Type-B. The procedure for generating the testing, training and validation sets is fully described in Procedure 1.

The procedure takes two variables as input: D and C. Variable D represents the dataset dimensions, while variable C holds the set of distinct attack classes or subclasses. The output is multiple subsets (i.e., training, testing and validation). The number of generated subsets depends on the size of variable C, which is the number of distinct attack classes or subclasses. In Lines 1-29, the procedure loops through each class or subclass in variable C. In Line 2, a copy of the original dataset is made. In Lines 3–5, the output subsets are initialised. In Lines 6–27, the procedure loops thorough each instance in the input dataset. In Lines 7–10, the procedure verifies if the attack class of the current instance matches the one in the outer loop, and if so, it will be added to the testing subset and removed from the temporary dataset. Lines 11–14 verify whether the attack subclass matches the subclass of the outer loop. In this case, the instance will be appended to the testing subset. Lines 15–26, confirm the problem type, namely binary classification and multi-class classification of Type-A or Type-B. In case it is a binary classification problem, both attack and sub-attack labels are dropped, and only the binary label is stored. Conversely, the binary and the sub-attack labels are dropped when it is a Type-A multi-class problem, whereas the binary and the attack labels are both dropped if it is a Type-B multi-class problem. Finally, in Line 28, a function is called to split the temporary dataset into training and validation subsets using the stratified random split method with 90% for training and 10% for validation. The function is based-on a pseudo-random function, which uses a fixed seed to ensure the repeatability.

Procedure 1 Generate testing, training and validation sets for unknown attacks.

```
Input: $D$, dataset of size $(m \times n)$
Input: $C \leftarrow []$, set of size $l$; Type-A or Type-B
Output: $l \times$Testing subsets of size $(m-1 \times n_1)$
Output: $l \times$Training subsets of size $(m-1 \times n_2)$
Output: $l \times$Validation subsets of size $(m-1 \times n_3)$

Ensure: $n_2 = (n - n_1) * 0.9$
Ensure: $n_3 = (n - n_1) * 0.1$
 1: for all $c \in C$ do
 2:     $D_{temp} \leftarrow D$
 3:     $D_{test^c} \leftarrow []$
 4:     $D_{train^{C-c}} \leftarrow []$
 5:     $D_{val^{C-c}} \leftarrow []$
 6:     for all $d \in D$ do
 7:         if Type-A and d.class == c then
 8:             $D_{test^c}$ += $d$
 9:             $D_{temp}$ -= $d$
10:         end if
11:         if Type-B and d.subclass == c then
12:             $D_{test^c}$ += $d$
13:             $D_{temp}$ -= $d$
14:         end if
15:         if BinaryClass then
16:             del $d.class$
17:             del $d.subclass$
18:         end if
19:         if MultiClass and Type-A then
20:             del $d.binary$
21:             del $d.subclass$
22:         end if
23:         if MultiClass and Type-B then
24:             del $d.binary$
25:             del $d.class$
26:         end if
27:     end for
28:     $D_{train^{C-c}}, D_{val^{C-c}}$ = StratifiedRandomSplit($D_{temp}$, 0.9, 1586512076128)
29: end for
30: Function StratifiedRandomSplit($l$, $r$, $s$):
31:     return $r$, $(1 - r)$ from $l$ using stratified random sampling with seed $s$
32: End Function
```

As previously described in Table 2, the UNSW-NB15 dataset has nine attack classes and no subclasses; therefore, only unknown attacks of Type-A were generated. For each of the nine attack classes, four models were built, trained and tested, two of which were shallow and deep binary classification models, and the other two were multi-class classification models, resulting in a total of 36 models. On the other hand, the Bot-IoT has both class and subclass labels; thus, both Type-A and Type-B testing sets were generated. Four Type-A attack classes and 10 Type-B attack subclasses were found; therefore, 16 binary classification models were built, in addition to 40 multi-class classification models. Overall, ninety-two models were used to evaluate the ability of shallow and deep ANN classifiers to detect unknown attacks under the proposed categorisation. Figure 2 illustrates the process of generating the subsets and the associated ML models.

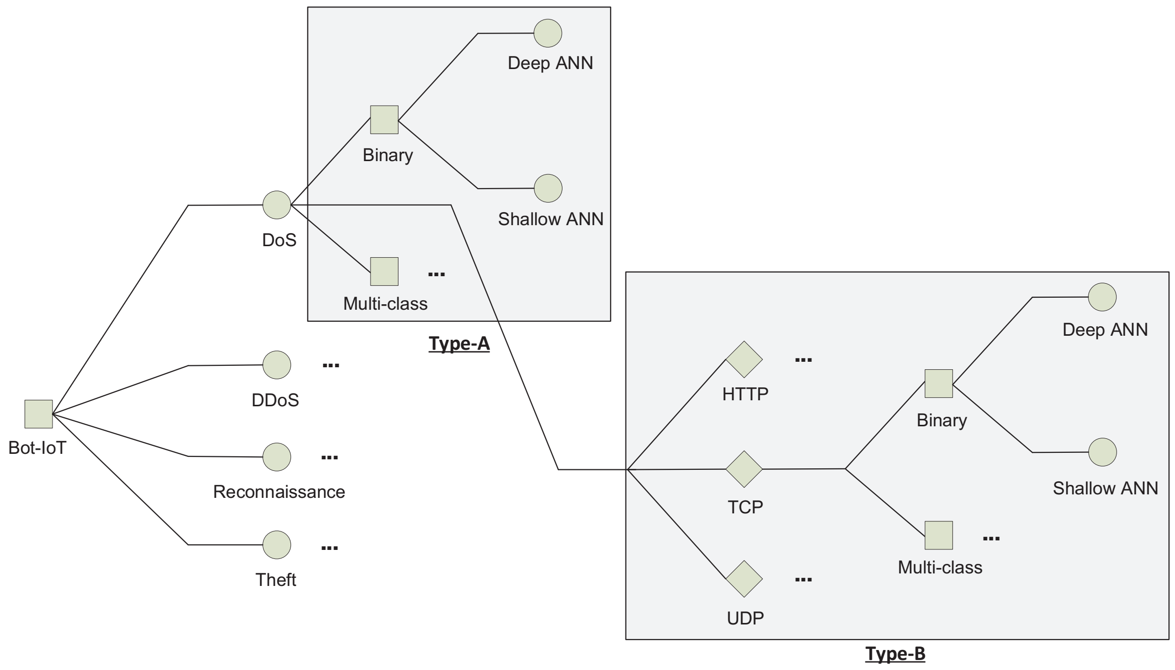

Figure 2. Example of Type-A and Type-B sets' generation and machine learning models' preparation.

5. Results

In this section, the results are presented and discussed for each dataset. Each model was trained, validated and tested 30 times, and the overall average value for all the runs is reported. This was done to ensure that the results were as accurate as possible. The accuracy, F1-score and classification error rate values were then used as the evaluation metrics to measure the model's ability to correctly classify unknown attacks. The accuracy in Equation (1) is an evaluation metric that measures the ratio of the correctly classified instances to the total number of instances. The F1-score in Equation (2) estimates the model's ability to correctly classify instances as a weighted average of the precision and recall. The classification error rate in Equation (3) measures the ratio of the misclassified instances to the total number of instances. A higher accuracy and F1 value imply better performance, while a higher classification error rate value implies poor performance.

$$Accuracy = \frac{TP + TF}{TP + TF + FP + FN} \tag{1}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2}$$

$$Classification\ error\ rate = 1 - \frac{TP + TF}{TP + TF + FP + FN} \tag{3}$$

Most importantly, the definition of “correctly classified” should be properly set and explained. For the binary classification models, the classification was considered correct (i.e., true) if the instance in the testing set was classified as an attack; otherwise, it was considered a misclassification (i.e., false). This is consistent with regular machine learning generalisation error measurements.

Conversely, when working with multi-class classification models, one cannot simply apply the same concept nor follow the norms of machine learning generalisation error measurements, because the class in the testing set is not present in the training set. Therefore, the model has never learned to classify the class in the first place, and thus, it is an unknown class/attack. Consequently, for multi-class Type-A models, if the instance is classified as benign, it is considered a misclassification; otherwise, it is considered correct. The logic behind this is that the classifier was able to correctly identify it as an attack, but placed it into an incorrect category because it was never trained to learn that class. This is analogous to a child who has just started learning about animals and who broadly knows cats, dogs, birds and fishes. Then, he/she is presented with a picture of a horse and asked to classify it. If he/she says that it is an animal, then surely his/her response could be construed as correct given that he/she does not know any better. On the other hand, with multi-class Type-B models, the predication is considered correct only if the model assigns the instance a subclass from within its parent class. The logic behind this is that the model was able to approximate the subclass to the closest correct one; thus, this is considered correct classification. Returning to the previous analogy, if the same child also gained knowledge about a few breeds of dog—for example, poodle and bulldog—and then was presented with a picture of a German shepherd, if he/she were able to classify it as a dog, then he/she would be considered to be correct; otherwise, his/her classification would be clearly incorrect.

In Table 5, the results of the binary classification models using the UNSW-NB15 dataset are summarised. They show that both the shallow and deep models were able to detect four out of the nine Type-A unknown attacks (i.e., Analysis, DoS, Exploits and Shellcode) with a marginal classification error rate and high accuracy and F1-score values, which is reasonable given that it is a binary classifier. On the other hand, the shallow classifiers were completely unable to detect three types of unknown attacks, namely Worms, Backdoors and Generic attacks, with a 100.00% classification error rate and zero accuracy and F1-score. When compared with their corresponding deep classifiers, both Worms and Backdoors attacks were detected with minimal error rate (i.e., 2.87% and 0.77%, respectively). The classification error rate for the Generic attacks class was still high (i.e., 68.87%) with poor accuracy (i.e., 31.13%) and a poor F1-score value (i.e., 47.48%) using the deep classifier, suggesting that for Backdoors and Worms classes, the deep models suffered from over-fitting, which is a common issue with ANN models. Interestingly, with regard to the Fuzzers and the Reconnaissance classes, the deep classifiers performed poorly in comparison with the shallow classifier, with classification error rates of 67.63% and 40.85% as opposed to 8.33% and 17.09%, respectively. It is clear from the experimental result that neither the deep nor the shallow models were able to detect all types of unknown attacks with high accuracy and F1-score values while maintaining a marginal classification error rate.

Table 5. Error rate for Type-A binary class models using the UNSW-NB15 dataset.

Class	**Shallow ANN**			**Deep ANN**		
	Accuracy	**F1-Score**	**Error Rate**	**Accuracy**	**F1-Score**	**Error Rate**
Analysis	0.9944	0.9972	0.0056	0.8771	0.9345	0.1229
Backdoors	0.0000	0.0000	1.0000	0.9923	0.9961	0.0077
DoS	0.9951	0.9975	0.0049	0.9710	0.9853	0.0290
Exploits	0.8786	0.9354	0.1214	0.8470	0.9172	0.1530
Fuzzers	0.9167	0.9566	0.0833	0.3237	0.4891	0.6763
Generic	0.0000	0.0000	1.0000	0.3113	0.4748	0.6887
Reconnaissance	0.8291	0.9065	0.1709	0.5915	0.7433	0.4085
Shellcode	0.9921	0.9960	0.0079	0.9576	0.9784	0.0424
Worms	0.0000	0.0000	1.0000	0.9713	0.9854	0.0287

In Table 6, the results of the multi-class classification models using the UNSW-NB15 dataset are shown, and they clearly indicate that the shallow classifiers were unable to detect all of the unknown attacks with a high classification error rate and low accuracy and F1-score values, except for the Worms attack, in which both accuracy and F1-score values were high (98.85% and 99.42, respectively) and the classification error rate was significantly low (i.e., 1.15%); this is as it should be, as shallow classifiers are known to yield good results with smaller datasets [47]. On the other hand, the deep multi-class classifiers were able to detect four out of the nine unknown attacks with a minimal classification error rate (<14%), namely Analysis, DoS, Shellcode and Worms attacks. As regards the remaining attacks, the classification error rate was considerably high (between 43–95%). Interestingly, the deep classifiers performed poorly in comparison with shallow classifiers under the Generic and the Worms attacks, with classification error rates of 76.67% and 9.77% as opposed to 43.35% and 1.15%, respectively.

Table 6. Error rate for Type-A multi-class models using the UNSW-NB15 dataset.

Class	**Shallow ANN**			**Deep ANN**		
	Accuracy	**F1-Score**	**Error Rate**	**Accuracy**	**F1-Score**	**Error Rate**
Analysis	0.0000	0.0000	1.0000	0.8741	0.9328	0.1259
Backdoors	0.0000	0.0000	1.0000	0.7952	0.8859	0.2048
DoS	0.0000	0.0000	1.0000	0.9338	0.9658	0.0662
Exploits	0.2779	0.4349	0.7221	0.5177	0.6822	0.4823
Fuzzers	0.0593	0.1119	0.9407	0.2643	0.4181	0.7357
Generic	0.5665	0.7233	0.4335	0.2333	0.3784	0.7667
Reconnaissance	0.2597	0.4123	0.7403	0.6811	0.8103	0.3189
Shellcode	0.2396	0.3865	0.7604	0.8643	0.9272	0.1357
Worms	0.9885	0.9942	0.0115	0.9023	0.9486	0.0977

From these results, one could conclude that shallow multi-class classifiers are ineffective at detecting unknown attacks, with an overall weighted average classification error rate of 55.57%. Conversely, although the deep multi-class classifiers were able to detect several unknown attacks with a marginal classification error rate and an overall weighted average classification error rate of 28.28%, the fact that several other unknown attacks were missed with considerably high classification error rates and low accuracy and F1-score means that they are not a suitable solution. Furthermore, because the shallow classifiers outperformed their respective deep classifiers in terms of accuracy and F1-score when detecting several unknown attacks, one cannot simply exclude shallow classifiers and only rely on deep classifiers.

The results for the Bot-IoT dataset using binary classification models under Type-A unknown attacks are summarised in Table 7. Both shallow and deep models showed a negligible classification error rate for the DDoS class (<6%). Conversely, for the remaining classes, the classification error rate ranged from 12.43% up to 85.97% with a weighted average of 20.60% for the shallow models and 49.94% for the deep classifiers. Comparing these results with those obtained using the UNSW-NB15 dataset, there was clearly a noticeable improvement in the overall classification error rate, accuracy and F1-score; however, when comparing the generalisation error metrics for the same classes (i.e., DoS and Reconnaissance) from both datasets, the classification error rate was considerably higher in both datasets.

Table 8 shows the results for the Bot-IoT dataset using the multi-class models for Type-A attacks. The shallow classifiers showed negligible classification error rates across all attack classes (<7%), high accuracy (>93%) and high F1-score (>96%), whilst the deep models showed considerable classification error rates (>30%) for all classes except DDoS (<7%). This can be explained by the fact that shallow models outperform deep classifiers with smaller datasets as ANN models tend to stick to local optima.

Table 7. Error rate for Type-A binary class models using the Bot-IoT dataset.

Class	**Shallow ANN**			**Deep ANN**		
	Accuracy	**F1-Score**	**Error Rate**	**Accuracy**	**F1-Score**	**Error Rate**
DoS	0.6983	0.8223	0.3017	0.7445	0.8535	0.2555
DDoS	0.9773	0.9885	0.0227	0.9474	0.9730	0.0526
Reconnaissance	0.6246	0.7689	0.3754	0.1403	0.2461	0.8597
Theft	0.8757	0.9337	0.1243	0.1703	0.2911	0.8297

Table 8. Error rate for Type-A multi-class models using the Bot-IoT dataset.

Class	**Shallow ANN**			**Deep ANN**		
	Accuracy	**F1-Score**	**Error Rate**	**Accuracy**	**F1-Score**	**Error Rate**
DoS	0.9983	0.9991	0.0017	0.6953	0.8203	0.3047
DDoS	0.9315	0.9645	0.0685	0.9307	0.9641	0.0693
Reconnaissance	1.0000	1.0000	0.0000	0.0000	0.0000	1.0000
Theft	1.0000	1.0000	0.0000	0.0536	0.1017	0.9464

For Type-B unknown attacks, the results were convergent for both binary and multi-class models with an overall weighted average classification error rate of around 25% for the shallow models and 22.5% for the deep models. Interestingly, DoS and DDoS classes showed the lowest error rates amongst all attack classes, except for the DDoS HTTP class, which showed the highest error rate under both shallow (>84%) and deep (>76%) classifiers. Tables 9 and 10 summarise the results for binary and multi-class Type-B unknown attacks using the Bot-IoT dataset, respectively.

Table 9. Error rate for Type-B binary class models using the Bot-IoT dataset.

Class	Shallow ANN			Deep ANN		
	Accuracy	F1-Score	Error Rate	Accuracy	F1-Score	Error Rate
Data Exfiltration	0.9573	0.9782	0.0427	0.9145	0.9554	0.0855
Keylogging	0.9864	0.9931	0.0136	0.2577	0.4098	0.7423
OS Fingerprint	0.9459	0.9722	0.0541	0.9808	0.9903	0.0192
Service Scan	0.8321	0.9084	0.1679	0.6376	0.7787	0.3624
DoS HTTP	0.9828	0.9913	0.0172	0.9582	0.9786	0.0418
DoS TCP	0.9806	0.9902	0.0194	0.9669	0.9832	0.0331
DoS UDP	0.9707	0.9851	0.0293	0.9934	0.9967	0.0066
DDoS HTTP	0.1504	0.2615	0.8496	0.2371	0.3833	0.7629
DDoS TCP	0.9875	0.9937	0.0125	0.9940	0.9970	0.0060
DDoS UDP	0.9848	0.9923	0.0152	0.9996	0.9998	0.0004

Table 10. Error rate for Type-B multi-class models using the Bot-IoT dataset.

Class	Shallow ANN			Deep ANN		
	Accuracy	F1-Score	Error Rate	Accuracy	F1-Score	Error Rate
Data Exfiltration	0.0000	0.0000	1.0000	0.9322	0.9649	0.0678
Keylogging	0.0000	0.0000	1.0000	0.0007	0.0014	0.9993
OS Fingerprint	0.0000	0.0000	1.0000	0.9918	0.9959	0.0082
Service Scan	0.1124	0.2021	0.8876	0.3266	0.4924	0.6734
DoS HTTP	1.0000	1.0000	0.0000	0.9075	0.9515	0.0925
DoS TCP	0.9999	0.9999	0.0001	0.9757	0.9877	0.0243
DoS UDP	0.9693	0.9844	0.0307	0.9995	0.9997	0.0005
DDoS HTTP	0.0001	0.0002	0.9999	0.0275	0.0534	0.9725
DDoS TCP	1.0000	1.0000	0.0000	0.9915	0.9957	0.0085
DDoS UDP	1.0000	1.0000	0.0000	0.9873	0.9936	0.0127

When taking into consideration the high overall weighted average classification error rate from all experiments (50.09%), it is obvious that neither shallow nor deep models can be considered a viable solution for the detection of unknown attacks under the proposed categorisation. This supports our research proposal, as the current common classifiers were not able to detect unknown attacks with an acceptable generalisation error metrics.

6. Conclusions

An intrusion detection system is an effective tool for detecting potential security threats and violations. Nonetheless, detecting unknown attacks has proven to be a challenging issue due to the constant growth of new attacks. Detecting unknown attacks is an open research area. The literature lacks a consistent definition of what constitutes an unknown attack. In this work, the authors proposed a new categorisation to define unknown attacks as Type-A of unknown attacks, which represents a completely new category of attacks, and Type-B, which represents unknown attacks within already known categories of attacks. Then, the authors provided an experimental evaluation using modern shallow and deep ANN models and two benchmark datasets that are commonly used in IDS research. A total of 92 different models were tested, and the results showed that they were incapable of detecting several classes of unknown attacks, as well as exhibiting a significant overall error rate of around 50%, which suggests the need for innovative approaches and techniques to address this problem. Machine learning (ML) classification algorithms, supervised and unsupervised alike, are used to handle problems with predefined classes of data at the model creation time. Whether the data are labelled or not, the number of overall classes, which represents the number of attack categories in our domain here, is fixed at the model creation time during the learning stage. Classifying, with high accuracy, the unknown security attacks that do not exist during the ML model creation requires a different approach.

As a future work, the authors are planning to:

- Evaluate the performance of hybrid supervised and unsupervised anomaly-based ML models in detecting unknowns attacks.
- Examine whether the lightweight ML classifiers are capable of detecting unknown attacks in the IoT environment.
- Study the ability of discovering an adversarial attack that might result in misled prediction as a type of unknown attack.

Author Contributions: Conceptualisation, M.A.-Z.; methodology, M.A.-Z.; validation, M.A.-Z., S.A. and M.A.; writing, original draft preparation, M.A.-Z.; writing, review and editing, M.A.-Z., S.A. and M.A.; supervision, S.A. and M.A. All authors read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Knuth, D. Store Stats for Mobile Apps. 2020. Available online: https://42matters.com/stats (accessed on 24 April 2020).
2. Nisioti, A.; Mylonas, A.; Yoo, P.D.; Katos, V. From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3369–3388. [CrossRef]
3. Tran, N.N.; Sarker, R.; Hu, J. An Approach for Host-Based Intrusion Detection System Design Using Convolutional Neural Network. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 116–126. [CrossRef]
4. Aliakbarisani, R.; Ghasemi, A.; Wu, S.F. A data-driven metric learning-based scheme for unsupervised network anomaly detection. *Comput. Electr. Eng.* **2019**, *73*, 71–83. [CrossRef]
5. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
6. Fernandes, G.; Rodrigues, J.J.P.C.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. *Telecommun. Syst.* **2018**, *70*, 447–489. [CrossRef]
7. Moustafa, N.; Hu, J.; Slay, J. A holistic review of Network Anomaly Detection Systems: A comprehensive survey. *J. Netw. Comput. Appl.* **2019**, *128*, 33–55. [CrossRef]
8. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
9. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
10. Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Estrada-Solano, F.; Caicedo, O.M. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J. Internet Serv. Appl.* **2018**, *9*. [CrossRef]
11. Jin, D.; Lu, Y.; Qin, J.; Cheng, Z.; Mao, Z. SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. *Comput. Secur.* **2020**, *97*, 101984. [CrossRef]
12. Kukielka, P.; Kotulski, Z. Analysis of neural networks usage for detection of a new attack in IDS. *Ann. UMCS Inform.* **2010**, *10*, 51–59. [CrossRef]
13. Kukielka, P.; Kotulski, Z. Adaptation of the neural network-based IDS to new attacks detection. *arXiv* **2010**, arXiv:1009.2406.
14. Kukielka, P.; Kotulski, Z. New Unknown Attack Detection with the Neural Network–Based IDS. In *The State of the Art in Intrusion Prevention and Detection*; Auerbach Publications: New York, NY, USA, 2013; pp. 259–284. [CrossRef]

15. Bao, D.; Goto, Y.; Cheng, J. Predicting New Attacks for Information Security. In *Computer Science and its Applications*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1353–1358. [CrossRef]
16. Bao, D.; Wagatsuma, K.; Gao, H.; Cheng, J. Predicting New Attacks: A Case Study in Security Analysis of Cryptographic Protocols. In *Lecture Notes in Electrical Engineering*; Springer: Singapore, 2016; pp. 263–270. [CrossRef]
17. Ajjouri, M.E.; Benhadou, S.; Medromi, H. LnaCBR:Case Based Reasoning Architecture for Intrusion Detection to Learning New Attacks. *Rev. MéDiterranéEnne Des TéLéCommunications* **2016**, *6*, 54–59.
18. Sellami, L.; Idoughi, D.; Tiako, P.F. Detection of New Attacks on Ubiquitous Services in Cloud Computing and Against Measure. *Adv. Sci. Lett.* **2016**, *22*, 3168–3172. [CrossRef]
19. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2017**, *25*, 152–160. [CrossRef]
20. Meira, J.; Andrade, R.; Praça, I.; Carneiro, J.; Marreiros, G. Comparative Results with Unsupervised Techniques in Cyber Attack Novelty Detection. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 103–112. [CrossRef]
21. Amato, F.; Moscato, F.; Xhafa, F.; Vivenzio, E. Smart Intrusion Detection with Expert Systems. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 148–159. [CrossRef]
22. Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. *IEEE Access* **2018**, *6*, 33789–33795. [CrossRef]
23. Santikellur, P.; Haque, T.; Al-Zewairi, M.; Chakraborty, R.S. Optimized Multi-Layer Hierarchical Network Intrusion Detection System with Genetic Algorithms. In Proceedings of the 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–7. [CrossRef]
24. ul Haq Qureshi, A.; Larijani, H.; Ahmad, J.; Mtetwa, N. A Heuristic Intrusion Detection System for Internet-of-Things (IoT). In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 86–98. [CrossRef]
25. Bajaj, K.; Arora, A. Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods. *Int. J. Comput. Appl.* **2013**, *76*, 5–11. [CrossRef]
26. Qureshi, A.U.H.; Larijani, H.; Mtetwa, N.; Javed, A.; Ahmad, J. RNN-ABC: A New Swarm Optimization Based Technique for Anomaly Detection. *Computers* **2019**, *8*, 59. [CrossRef]
27. Khare, N.; Devan, P.; Chowdhary, C.; Bhattacharya, S.; Singh, G.; Singh, S.; Yoon, B. SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection. *Electronics* **2020**, *9*, 692. [CrossRef]
28. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine. *Electronics* **2020**, *9*, 173. [CrossRef]
29. Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.; Ghogho, M.; Moussa, F.E. DeepIDS: Deep Learning Approach for Intrusion Detection in Software Defined Networking. *Electronics* **2020**, *9*, 1533. [CrossRef]
30. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. *Electronics* **2020**, *9*, 916. [CrossRef]
31. Jo, W.; Kim, S.; Lee, C.; Shon, T. Packet Preprocessing in CNN-Based Network Intrusion Detection System. *Electronics* **2020**, *9*, 1151. [CrossRef]
32. Ghafir, I.; Prenosil, V. Proposed Approach for Targeted Attacks Detection. In *Lecture Notes in Electrical Engineering*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 73–80. [CrossRef]
33. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* **2019**, *2*. [CrossRef]
34. Passeri, P. Cyber Attacks Statistics, 2020. Available online: https://www.hackmageddon.com/category/security/cyber-attacks-statistics (accessed on 23 September 2020)

35. Mawgoud, A.A.; Taha, M.H.N.; Khalifa, N.E.M.; Loey, M. Cyber Security Risks in MENA Region: Threats, Challenges and Countermeasures. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 912–921. [CrossRef]
36. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
37. Bay, S.D.; Kibler, D.F.; Pazzani, M.J.; Smyth, P. The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation. *SIGKDD Explor.* **2000**, *2*, 81. [CrossRef]
38. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the CISDA'09 Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada , 8–10 July 2009; pp. 53–58.
39. Sangster, B.; O'Connor, T.J.; Cook, T.; Fanelli, R.; Dean, E.; Adams, W.J.; Morrell, C.; Conti, G. Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets. In *Proceedings of the 2nd Conference on Cyber Security Experimentation and Test*; USENIX Association: Berkeley, CA, USA, 2009; p. 9.
40. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
41. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
42. 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 Systems Security and Privacy. SCITEPRESS-Science and Technology Publications, Madeira, Portugal, 22–24 January 2018; pp. 108–116. [CrossRef]
43. Haider, W.; Hu, J.; Slay, J.; Turnbull, B.; Xie, Y. Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *J. Netw. Comput. Appl.* **2017**, *87*, 185–192. [CrossRef]
44. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
45. Parmisano, A.; Garcia, S.; Erquiaga, M.J. *A Labeled Dataset with Malicious and Benign IoT Network Traffic*; Stratosphere Laboratory: Praha, Czech Republic, 2020.
46. Al-Zewairi, M.; Almajali, S.; Awajan, A. Experimental Evaluation of a Multi-layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 11–13 October 2017; pp. 167–172. [CrossRef]
47. Pasupa, K.; Sunhem, W. A comparison between shallow and deep architecture classifiers on small dataset. In Proceedings of the 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 5–6 October 2016; pp. 390–395. [CrossRef]
48. Hodo, E.; Bellekens, X.J.A.; Hamilton, A.W.; Tachtatzis, C.; Atkinson, R.C. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. *arXiv* **2017**, arXiv:1701.02145.
49. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
50. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl. Based Syst.* **2020**, *189*, 105124. [CrossRef]