Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DISCOVERING UNKNOWN BOTNET ATTACKS ON IOT DEVICES USING SUPERVISED SHALLOW AND DEEP LEARNING CLASSIFIERS

[1]MALEK AL-ZEWAIRI, [1]SUFYAN ALMAJALI, [2]MOUSSA AYYASH

[1]Department of Computer Science, Princess Sumaya University for Technology, Amman 11941, Jordan

[2]Department of Computing, Information, and Mathematical Sciences and Technology, Chicago State University, Chicago, IL, 60628, USA

E-mail: [1]m.alzewairi@jisdf.org, [1]s.almajali@psut.edu.jo, [2]msma@ieee.org

ABSTRACT

Computer networks constitute the vital artery of the information and communications technology era, allowing heterogeneous devices to communicate and share data. The immense number of Internet-connected devices with unpatched security vulnerabilities makes them susceptible to massive security attacks. Detecting unknown security attacks continues to be a major challenge, as they have been ranked constantly in the top three attack techniques since 2014. In this paper, the researchers aim to study the ability of supervised shallow and deep learning classifiers in detecting unknown botnet attacks on IoT devices. The performance of shallow and deep supervised learning classifiers was studied and compared using a well-known dataset (i.e., the Aposemat IoT-23 dataset). A thorough and extensive experimentation process was conducted (1000 experiments in total were performed), in which 12 unknown attack types and 38 unknown attack subtypes were studied under binary and multiclass classification problem. The results showed that the overall weighted average classification error rate was considerably high (61.46–86.40%), which dictates the importance of finding novel approaches and techniques to detect unknown attacks.

Keywords: *Botnet, Deep Learning, IDS, IoT, IoT-23 Dataset, Unknown Attacks*

1. INTRODUCTION

The Internet of Things (IoT) revolution allowed commodity hardware to communicate with each other and exchange data over the Internet cheaply and effectively [1]. In 2018, the number of connected devices, whether traditional or IoT, surpassed 31 billion devices and several studies forecasted this number will exceed 50 billion by 2020 and reach 125 billion by 2030 [2]. The immense growth of IoT devices enabled IoT to be used in many applications, including but not limited to; home automation, connected autonomous vehicle (CAV), smart cities, industrial control systems (ICS), and the Internet of healthcare things (IoHT) [3]. This integration of IoT devices in several applications has widened the attack surface on the Internet and created significant cybersecurity threats, where IoT devices have been used to launch security attacks against critical infrastructure and key resources (CIKR) and Supervisory Control and Data Acquisition (SCADA) [4].

IoT-based attacks have significantly increased in the past 5-years. It is predicted to make up 25% of all the cybersecurity attacks on enterprises in 2020 [5]. Denial of service (DoS) and distributed denial of service (DDoS), malware, botnet, man-in-the-middle (MiTM), unauthorised access, sinkhole and wormhole are examples of common security threats on IoT devices [4], [6]. A botnet is a network of compromised computing devices, often referred to as Bots or Zombies, thus botnet. Botnets are controlled by an attacker or a group of attackers via a set of command and control (C&C) servers in a well-designed hierarchical order that offers redundancy and anonymity [7]. Attackers target vulnerable IoT devices to make them part of bigger botnets and use them to launch several types of attacks, such as DDoS [8].

An intrusion detection system (IDS) is one of the primary security controls that has been used for decades to inspect the network traffic and system applications for signs of intrusive activities [9]. Several studies have researched the various designs, methods and techniques of detecting network attacks on IoT devices using IDS [4], [6], [9-11]. A significant challenge for machine learning (ML)

based IDS is to detect unknown attacks that it has never encountered before during training [12].

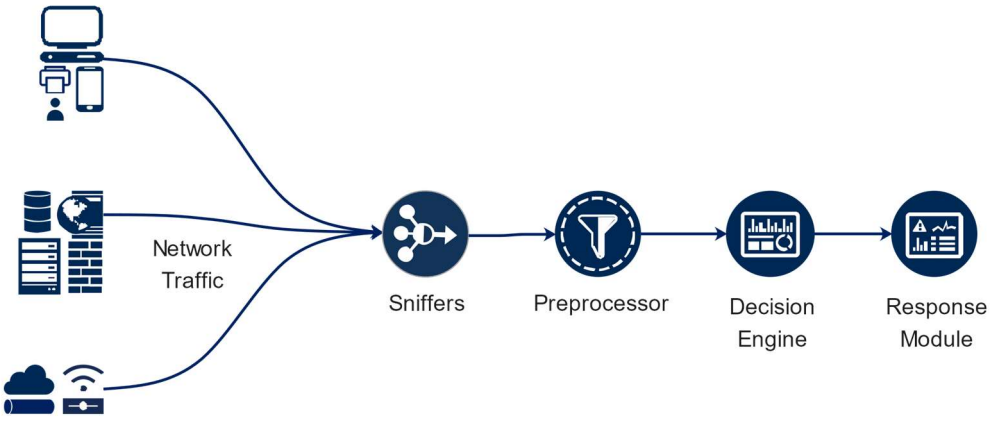


Figure 1: Simplified Intrusion Detection System Architecture.

The ideal IDS for resource-constrained networks, such as IoT, should not introduce any additional overhead to the existing infrastructure yet provide a high detection rate using lightweight design [10].

In this paper, the authors investigated the problem of detecting unknown botnet attacks targeting IoT devices using the notion of Type-A and Type-B unknown attacks presented in [10], where type-A refers to unknown attack type, and type-B refers to unknown attack subtypes. We compared shallow and deep learning (DL) classifiers ability to detect unknown botnet attacks on IoT devices in a supervised learning environment using the IoT-23 dataset [11] to train and test the classifiers. The authors approached the research problem as a binary class classification problem and a multiclass classification problem.

The rest of this paper is structured as follows: Section 2 discusses the different types of intrusion detection systems. Section 3 summarises the current studies in the literature related to the research problem. Section 4 explores the IoT-23 dataset selection process and the various preprocessing steps to prepare the dataset for analysis. Section 5 explains the comparative analysis approach used to study the research problem, then presents and discusses the results. Finally, in Section 6, the paper is concluded, and the future work is stated.

2. INTRUSION DETECTION SYSTEM (IDS)

Typically, the IDS comprises four key components, as shown in Figure 1: sniffer, pre-processor, decision engine and response module [9]. The sniffer collects raw data from various devices, which are then fed to the preprocessing module to extract a set of features to be used by the decision engine. For which, the decision engine analyse and classify the data into benign or malicious data based on any combination of pre-defined signatures of known malicious data, abnormalities in a protocol/standard or variation from a baseline profile for what is considered normal behaviour (i.e., signature-based, specification-based and anomaly-based, respectively). Finally, the response module applies the action determined by the decision engine (e.g., alert, block or ignore).

Several studies classify intrusion detection systems based on various taxonomies [6], [9], [12], [13]. Figure 2 shows the general taxonomy of IDS that includes six main categories of IDS classifications, which are: data source, detection technique, deployment architecture, deployed applications, anomaly type, and defence mechanism.

Data source, also referred to as monitored environment and monitored platform, classifies IDSs into three subcategories: network-based IDS (NIDS), host-based IDS (HIDS) and hybrid IDS. In NIDS, the sniffers monitor the network traffic. Conversely, the sniffer in HIDS monitors the activities and events on the host OS. Consequently, a hybrid IDS combines both functionalities of NIDS and HIDS.

With the detection technique, the IDS is categorised based on its approach to determining if the activities are benign or malicious. Four main methods are used to classify IDSs based on their detection technique: signature-based, anomaly-based, specification-based, and hybrid.

Signature-based IDS, commonly referred to as misuse-based, relies on a dataset of pre-defined signatures of malicious activities. The main advantage of this approach is the high accuracy and low false

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

alarm rate (FPR) in detecting known attacks. However, it fails in detecting unknown attacks or variants of known attacks. Conversely, an anomaly-based IDS works by building a baseline profile for normal behaviour and marks any variation as an attack using a pre-defined threshold. With this approach, the IDS can detect both known and unknown attacks. However, it suffers from high FPR and false-negative alarm rates.

A specification-based IDS works by studying how a protocol (e.g., HTTP) should work and identify any abnormality in the protocol/standard as an attack. This approach can detect both known and unknown attacks at a low FPR. However, it is very complex to design and operate. Moreover, not all systems implement the protocols equally.

Finally, hybrid IDS combines two or more detection techniques to overcome each method's limitations. For example, combining signature-based with anomaly-based detection can achieve high accuracy and low FPR rates while being able to detect unknown attacks.

In deployment architecture, the IDS is categorised into either centralised or distributed IDS based on the deployment architecture. A centralised IDS is often found in unified threat management (UTM) security solutions and is deployed at a single location, usually at the network gateway. This architecture is suitable for small and medium-size organisations (SMEs). A distributed IDS is deployed at multiple network segments and reports anomalies to a central management device where the decision engine is located. This architecture provides more visibility into anomalies and is often found in highly segregated networks.

IDS can also be categorised based on the specific application it is designed to protect. A backbone-based IDS is designed to inspect ingress and egress traffic on different network segments. It is often implemented at the network backbones in either centralised or distributed architecture. Conversely, a datacentre-based IDS is designed to protect data centres and is often utilised by organisations that rely on virtualisation technologies.

An access point-based IDS is usually found in a wireless intrusion detection system (WIDS), where the IDS is implemented at the access point level. An IoT-based IDS is designed to protect IoT applications such as wireless sensor networks (WSN) and IoHT.

A cloud-based IDS is designed to detect internal threats in cloud computing environments where traditional IDS are incapable of providing the necessary protection. It is deployed at the cloud service provider (CSP) level to protect its internal infrastructure and offer it as a SaaS (Software as a Service) for its customers to protect their platforms.

Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

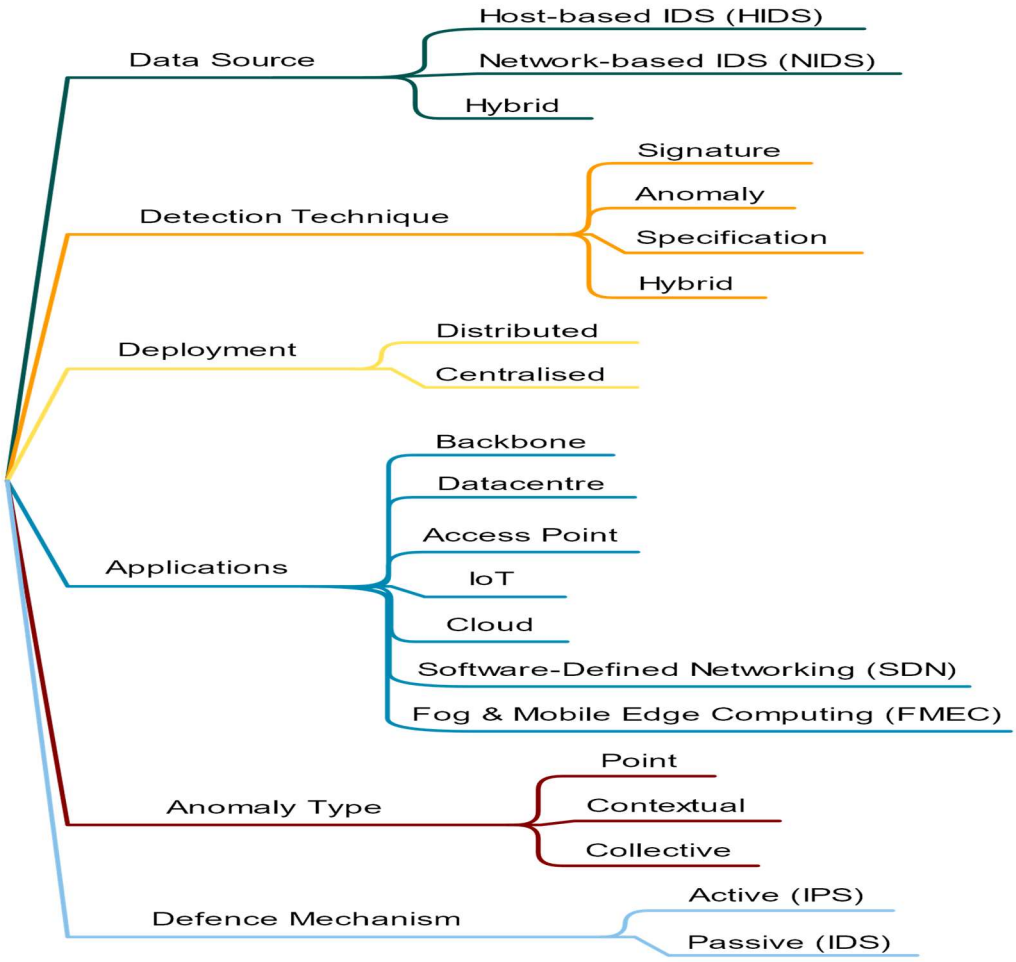


Figure 2: Taxonomy Of The Intrusion Detection System.

An SDN-based IDS is designed to fit the software-defined network architecture where the data plane is separated from the control plane. It is often perpetrated as software implemented at the controller level. While a fog and mobile edge computing (FMEC) based IDS is implemented in a decentralised architecture at the network edge to accommodate FMEC networks decentralised nature.

Another method to categorise anomaly-based IDS is based on the variation of the normal activities that can be used to distinguish between benign and malicious events. The anomaly type is tightly related to the detection approach, and there are three main anomaly types: point, contextual, and collective anomalies.

Point anomaly, also referred to as Outlier, occurs when some of the data are significantly different from the average. Figure 3 (a) shows an example of a point anomaly.

The contextual anomaly occurs when an event is anomalous in a specific context while normal in another. For example, in Figure 3 (b), the five blue points with a Y-axis value of 0.8 forming a valley are not considered outliers because the context indicates a continuous decreased line. However, the single orange point with the same Y-axis value is an outlier because all the neighbouring points have significantly higher Y-axis values.

On the other hand, collective anomaly looks at a collection of events that act anomalously with respect to the entire data pattern, as shown in Figure 3 (c).

Finally, with Defence Mechanism, an IDS is categorised into either an Active or a Passive IDS based on its response after detecting an attack. An active IDS, often referred to as Intrusion Prevention System (IPS), responds to security alerts by automatically blocking potential malicious activities. While with a passive IDS, the decision to deal with possible malicious activities is made manually by a human (i.e., block or allow), and the IDS only automates the alerting process.

3. LITERATURE REVIEW

Over the past decade, the research community has been actively working on intrusion detection solutions suitable for the resource-constrained IoT environment. Several survey studies were conducted [3], [4], [6], [14]–[22]. The researchers agreed on the importance of studying ML techniques to address the IDS special needs in IoT and its importance on protocol-level attacks.

In [7], the authors proposed a hybrid anomaly-based IDS architecture for detecting botnet attacks in IoT networks. The proposed architecture extends the authors' previous work [23], [24] by introducing a feature selection subsystem to make the architecture lightweight. Mainly, the architecture consists of two parts: A Model Builder and an Attack Detector. The model builder is responsible for building, training, and selecting ML models to be used by the attack detector to identify botnet attacks; where, each model is trained to detect one type of botnet attacks (i.e., the dataset has one type of attacks in addition to the benign instances). Then, the model with the highest accuracy is selected by the attack detector. The authors utilised the N-BaIoT dataset [8] and compared their results with three other ML algorithms; namely, NB, J48 and ANN. The results suggested that their hybrid method achieved comparable results with ANN and J48; while surpassed those of NB. The authors plan to extend their architecture to detect new types of unknown attacks.

In [25], the authors presented a two-stage cross-layer IDS to detect attacks on mobile ad-hoc networks (MANET) and WSN. The proposed architecture employs a two-layers heuristic detection approach based on the accumulated measure of fluctuation and linear regression for classification. It was evaluated against several attack scenarios, including blackhole and DDoS. The result showed a high detection rate with the F1 score ranged between 93% and 99.36%; however, the FPR was noticeably high (1.3-12%).

In [26], the authors highlighted the absence of proper software engineering (SE) process in developing IDS solutions in general and proposed three SE processes for designing efficient IDS for WSN. The results showed that following a proper SE process significantly improved the lifespan and energy consumption in WSN.

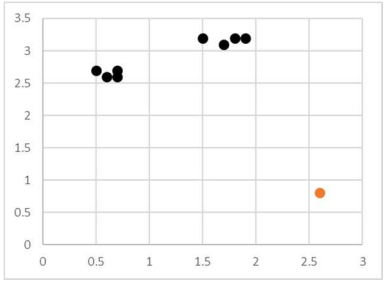
In [27], the authors proposed a real-time IDS for SCADA systems to detect attacks against the distributed network protocol (DNP3). Several ML algorithms were studied to classify the traffic in an ensemble approach.

In [28], the authors proposed a physically unclonable function (PUF) to identify IoT devices to replace cryptographic-based authentication, which has several applications in IoT-based IDS. Experimental tests showed that the proposed PUF has very high accuracy.

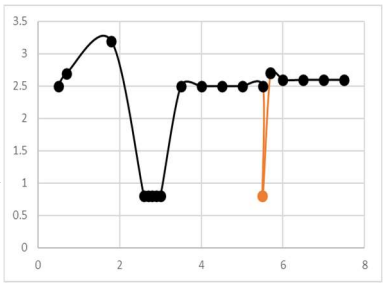
In [29], the authors studied the impact of different dimensionality reduction techniques on the IDS detection rate in IoT. Three methods (i.e., principal component analysis (PCA), random forest (RF), and filter-based dimensionality reduction) were studied. The results showed that RF has the best reduction to detection ratio. Moreover, the impact of a proposed ensemble-based dimensionality reduction method on IDS effectiveness for IoT was studied. Four different datasets were utilized in the study, and the results showed that the proposed method reduced the dimensionality of the datasets with 66% confidence.

In [30], the authors proposed a hybrid approach to detect known and unknown attacks in IoT environments using a three-phased IDS approach. The proposed IDS utilised signature-based techniques to detect known attacks and deep reinforcement learning to classify unknown attacks into four attack categories: DoS, probe, user-to-root (U2R) and remote-to-local (R2L). It was tested using the NSL-KDD dataset.

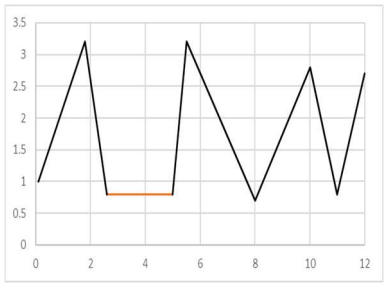
In [31], the authors highlighted the weakness of single classifier IDS and proposed a hybrid binary and multiclass ensemble IDS. The proposed IDS utilised 3 ML algorithms (i.e., J48, Random Forest, and Reptree), and it was evaluated using the KDD99 and the NSLKDD datasets. The



(a) Point Anomaly (Outlier)



(b) Contextual Anomaly



(c) Collective Anomaly

Figure 3: Examples of the different anomaly types.

results showed high detection rates (98.6-99%) with negligible RAF (0-0.5%).

In [32], the authors proposed a multi-stage IDS method for detecting attacks on IoT devices using a hybrid approach that combines supervised and unsupervised ML algorithms. The results showed a significant improvement in the accuracy (0.9842) compared to other traditional techniques such as NB and SVM.

4. IOT-23 DATASET

In machine learning, dataset selection constitutes the first and the most crucial step in building an ML model. It greatly affects the model training, validation, and performance evaluation process. Using a biased dataset would produce influenced results; simultaneously, using a dataset with poor data quality will yield weak models, thus, poor results. In anomaly-based IDS, the dataset has a vital role in determining the model prediction performance. A features-rich dataset with an accurate representation of benign and malicious activities would often produce better models.

The research community, over the years, has generated and critiqued numerous datasets for testing the various types of IDS models [11], [33]–[39]. The fact that not all datasets are created equally to fit all research questions, having the ML model training on a domain-specific dataset has a better chance of yielding better results [40].

In this research, the authors utilised the IoT dataset by the Stratosphere Laboratory (i.e., the IoT-23 [11]) to evaluate deep and shallow DL classifiers ability to detect unknown botnet attacks of type-A and type-B in an IoT network environment.

Originally, the dataset comprised more than 300 million records divided into 23 comma-separated values (CSV) files -hence the name- with each file contains a specific type of attacks and has 17 features and two class labels: a binary and a multiclass label, where the majority of the traffic was malicious. Table 1 shows the distribution for the binary class label. It shows a highly unbalanced dataset, where most instances are from the malicious class label.

Table 1: Binary class label distribution.

Class Label	Instances	Ratio
Benign	30,860,691	9.5%
Malicious	294,449,255	90.5%

After selecting a suitable dataset, the Feature Engineering stage commences. Its main objective is to extract useful information from raw data. It starts with the Feature Generation step, where raw data are captured and representative features are extracted to characterise it. Then, those features are studied, and a subset from them is chosen to train the ML model based on their ability to represent the dataset truly (e.g., information gain, information bias, Gini index, gain ratio, and entropy). This step is called Feature Selection. It is followed by the Feature Conversion step, where the data undergoes certain transformations to make its

format usable by the ML algorithm. Some ML algorithms perform better with scaled data (e.g., gradient descent-based algorithms), and in the feature normalisation step, the data are scaled up or down to fit a specific interval.

Typically, building the ML model comprises three steps; training, validation and testing. The ML model learns how to distinguish between benign and malicious activities in the training phase for an anomaly-based IDS. While in the validation phase, the model hyperparameters are tuned using an unbiased sample from the dataset. Finally, the model performance is estimated in the testing phase. Figure 4 summarises the stages associated with building an anomaly-based IDS.

After selecting the IoT-23 dataset, several preprocessing steps were taken. First, a new feature was generated to indicate the type-A attack each row represents, each CSV file holds. The existed multiclass label was considered as the type-B attack.

After that, five features were eliminated to avoid bias: source and destination IP (id.orig_p and id.resp_p), traffic origin indicator (local_orig and local_resp because their values are empty) and tunnel id (tunnel_parents since traffic with tunnel id is benign).

Then, all the 23 CSV files were merged into a single file, and all repeated instances were removed, which reduced the dataset size by 36.45 times. Thereafter, the string features (protocol, service, connection state and connection history) were converted to numerical values and normalised to [0-1] interval using the Min-Max method. Tables 2, 3 and 4 show the different classes in the IoT-23 dataset after the preprocessing stage. Comparing the binary class distribution before and after preprocessing (Table 2 and 3, respectively) shows how preprocessing helped balance benign and malicious instances.

The procedure described in [10] for generating training, validation and testing sets of unknown attacks was followed to create the binary and the multiclass subsets of type-A and type-B unknown attacks used in this study. All instances of each class label were removed from the training set to create a testing set of type-A unknown attack of that class. This procedure was done recursively for all type-A class labels shown in Table 3. Thus, generating 24 different testing sets (i.e., 12 binary and 12 multiclass). Similarly, the same procedure was followed using the subclass labels shown in Table 4 to generate type-B unknown attacks.

Table 2: Binary class distribution after pre-processing.

Class Label	Instances	Ratio
Benign	4,299,821	48.18%
Malicious	4,623,937	51.82%

Table 3: Type-A labels distribution after pre-processing.

Class Label	Instances	Ratio
Gagfyt	65,619	1.4191%
Hakai	4,112	0.0889%
HideAndSeek	266,956	5.7733%
IRCBot	30,343	0.6562%
Kenjiro	18,532	0.4008%
Linux.Hajime	131,477	2.8434%
Linux.Mirai	56,792	1.2282%
Mirai	3,919,807	84.7721%
Muhstik	113,959	2.4645%
Okiru	16,311	0.3528%
Torii	23	0.0005%
Trojan	6	0.0001%

In total, 76 testing sets were generated for type-B unknown attacks (i.e., 38 binary and 38 multiclass). Following the procedure proposed in [10], it made sure that the models are being tested on actual unknown attacks rather than unseen data, which are not truly unknown attacks.

Table 4: Type-B labels distribution after preprocessing.

Class Label	Subclass Label	Instances
Gagfyt	C&C-HeartBeat	95
	DDoS	65,524
Hakai	C&C	4,112
HideAndSeek	C&C	1
	PartOfAHorizontalPortScan	266,955
IRCBot	Attack	677
	C&C	1423
	PartOfAHorizontalPortScan	28,243
Kenjiro	Attack	4
	C&C-HeartBeat	9,947
	DDoS	1,094
	Okiru	3,004
	PartOfAHorizontalPortScan	4,478
	PartOfAHorizontalPortScan-Attack	5
Linux.Hajime	PartOfAHorizontalPortScan	131,477
Linux.Mirai	C&C-HeartBeat	5,777
	DDoS	39,584
	Okiru	11,431
Mirai	Attack	2,755
	C&C	12,093
	C&C-FileDownload	50
	C&C-HeartBeat-Attack	834

	C&C-HeartBeat-FileDownload	11
	C&C-Mirai	2
	C&C-PartOfAHorizontalPortScan	795
	DDoS	2,076,658
	FileDownload	15
	Okiru	227
	PartOfAHorizontalPortScan	1,826,367

Muhstik	Attack	5,962
	C&C	8
	PartOfAHorizontalPortScan	107,989
Okiru	C&C-HeartBeat	15,687
	Okiru	621
	Okiru-Attack	3
Torii	C&C-Torii	23
Trojan	C&C-FileDownload	3
	FileDownload	3

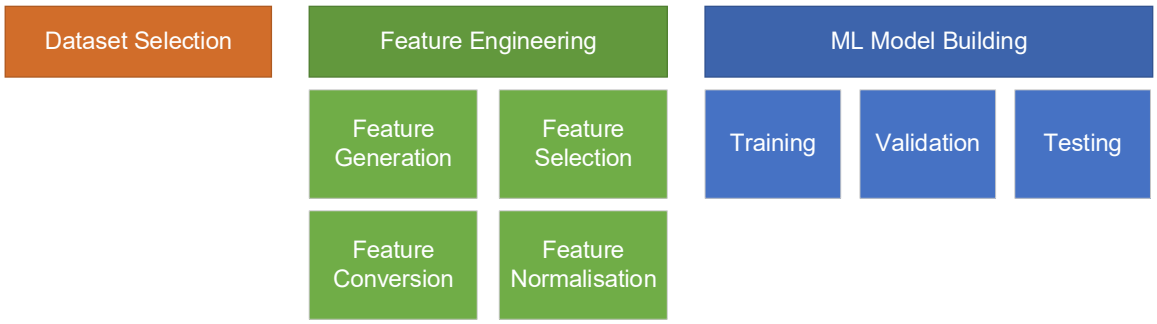


Figure 4: Stages of building an anomaly-based IDS.

5. COMPARATIVE ANALYSIS AND RESULTS

In this study, 1000 experiments were conducted to evaluate the performance of shallow and deep ML models in detecting unknown attacks related to Botnets in the IoT environment. In total, 200 ML models were built, trained and tested using the same methodology proposed in [10]. Type-A and type-B unknown attacks were tested as binary and multiclass classification problems to provide comprehensive analysis results. Each model was trained and tested five times, and the average results were reported.

The study refers to the shallow model as a DL classifier with a single hidden layer and ten neurons. In contrast, the Deep DL model consists of 5 hidden layers and ten neurons per layer, similar to those proposed in [10], [41]–[43]. Only the number of hidden layers is different between shallow and deep models. Table 5 summarises the DL model hyperparameters configurations. Figure 5 (a) and (b) visually illustrates binary shallow and deep models.

Table 5: DL models hyperparameters configurations.

Hyperparameter	Value
Activation function	Rectifier
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	5
Seed	1,586,512,076,128

Each of the 200 models was trained and validated using k-fold cross-validation using the stratified assignment and a fixed random seed.

Four generalisation error measures were utilised to evaluate each model performance: accuracy, recall, F1-score, and classification error rate. Accuracy is used to calculate the number of correctly classified instances concerning the total dataset size. On the other hand, recall measures the number of positively identified instances within a class. F1-Score, or simply the F measure, provides an insight into the overall model's accuracy. Finally, the classification error rate calculates the

fraction of misclassified instances in the dataset. Generally, a model with high accuracy, recall, F1-score and a low classification error rate is preferred.

Equations (1) to (4) explains how the accuracy, recall, F1-score and error rate are calculated, respectively.

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

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

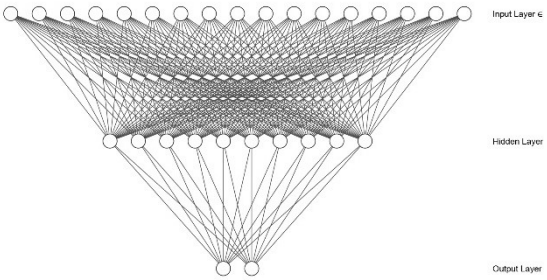
$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (3)$$

$$ErrorRate = 1 - \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

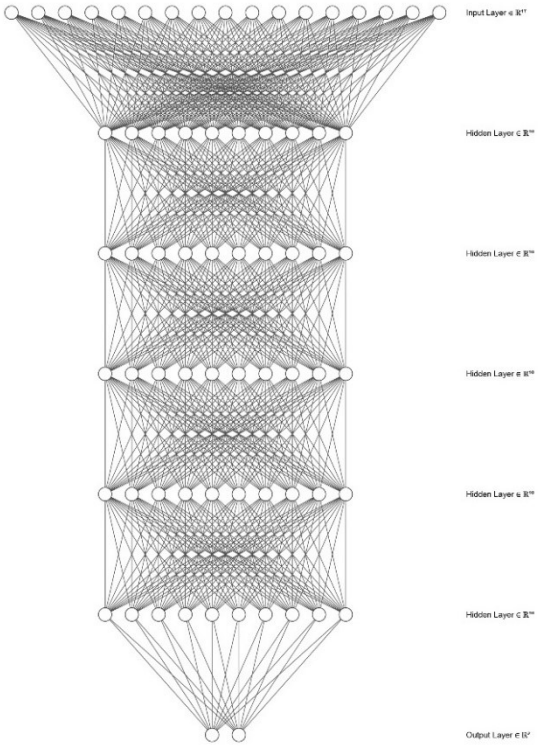
The results were split into four sections: Section 5.1 and 5.2 discuss the results of type-A and type-B unknown attacks. In contrast, the overall results were discussed in Section 5.3. Finally, open research issues and limitations of current work were presented in Section 5.4.

5.1. Results of Type-A Unknown Attacks

A comparison between shallow and deep DL classifiers in detecting type-A unknown attacks is presented in this section. The results are compared, first under a binary classification problem then under a multiclass classification problem.



(a)



(b)

Figure 5: Visualisation of the shallow and deep binary DL model

When examining the result for the shallow and deep classifiers, five of the type-A unknown attacks were discovered with marginal classification error (i.e., <0.10): Gagfyt, IRCBot, Linux.Hajime, Okiru, and Trojan. In the case of the Gagfyt attack, the four classifiers (i.e., shallow binary, shallow multiclass, deep binary and deep multiclass) were able to detect the attack at a near-zero error rate (i.e., <0.0017).

Conversely, seven out of the twelve type-A unknown attacks showed extremely high classification error rate under the four classifiers, namely, Hakai (0.34 – 0.50), HideAndSeek (0.44 – 0.74), Kenjiro (0.19 – 0.55), Linux.Mirai (0.89 – 0.95), Mirai (0.94 – 0.96), Muhstik (0.54 – 0.76), and Torii (0.26 – 0.65). It is was noticeable that shallow models, in general, have the highest error rate in comparison to the models and in particular as multiclass classifiers with two exceptions, the HideAndSeek and the Kenjiro attacks, where they outperformed their counterparts, DL models.

Figure 6 visually compares the classification error rate for the four models. Tables 6 and 7 summarise the results of detecting unknown

attacks of type-A, showing each model accuracy, recall, F1-score and classification error rate.

5.2. Results of Type-B Unknown Attacks

This section presents a comparison between shallow and deep DL classifiers in detecting type-B unknown attacks under both binary and multiclass classification problems. In total, there were 38 type-B attacks studied.

Unlike type-A unknown attacks, only three type-B unknown attacks were entirely detected by the four classifiers, namely, the PartOfAHorizontalPortScan subtype of IRCBot and Kenjiro attacks in addition to the CC-HeartBeat-FileDownload subtype of the Mirai botnet.

Even though the four models entirely discovered IRCBot attacks as a type-A attack, two sub-attacks (i.e., attack and cc) were missed with a significant error rate up to 0.41 when the models were training on detecting type-B unknown attacks. The Gagfyt attack was also missed by the four models with a high error rate (0.46 – 1.0).

On the contrary, the "Linux.Hajime" attack was detected with a near-zero error rate (0.0019) except by the deep model when trained to detect multiclass attacks where the error rate was marginally noticeable (0.13).

On the other hand, the deep binary model was able to completely detect one of the three subtypes of the Okiru attack (i.e., CC-HeartBeat), while its shallow counterpart model could detect it with a marginal error rate (i.e., 0.08).

As for the Trojan attack, the error rate for the multiclass shallow and deep models was around 100%; however, both binary classifiers could detect it with zero error rate. This was also true for the Troii CC-Filedownload attack.

Noticeably, the DDoS sub-attack from "Linux.Mirai" attack and the CC subtype of HideAndSeek were both entirely missed by the four classifiers. Simultaneously, the error rate of detecting the DDoS from the Mirai attack ranged between 0.94 and 0.97 by the four classifiers.

Tables 8 and 9 describe the results of detecting unknown attacks of type-B, showing each of the DL models' accuracy, recall, F1-score and classification error rate. Figure 7 shows a comparison between the classification error rate of the four DL models.

5.3. Overall Results

When considering the overall performance of each DL model, the overall weighted classification error average ranged between 0.85 and 0.87 for the shallow models under the type-A problem. In comparison, being between 0.87 and 0.88 for the deep classifiers under the same problem.

As for the models' performance under the type-B problem, the weighted classification error average slightly increased (0.88–0.89) for the shallow classifiers. However, while the weighted classification error average for deep binary classifiers decreased somewhat to become 0.86, it rose for the deep multiclass models to become 0.89.

Nonetheless, when the overall average classification error rate is calculated per attack type for type-A and type-B, the results indicated that the DL models for type-A outperformed those for type-B except in detecting Mirai attacks. This is because all models failed to classify the horizontal port scanning attack in both flavours of the Mirai attack (i.e., "Linux.Mirai" and Mirai).

Moreover, the performance of the type-B models in detecting the Gagfyt botnets worsened significantly with a 0.71 decrease. The Okiru and Trojan-based attacks also witnessed a similar decline in performance in terms of 0.47 and 0.46.

Overall, the evaluation results showed that some unknown attacks are better discovered using a shallow model (e.g., Gagfyt, HideAndSeek, and Kenjiro), while other attacks are detected the best using deep models (e.g., Hakai, Muhstik and Okiru). Moreover, some unknown attacks are best detected using a type-A DL model, such as Hakai, HideAndSeek, and Okiru. While some attacks are best discovered using type-B classifiers such as the Mirai-based attacks.

More importantly, no one classifier could detect all types of unknown attacks with an acceptable classification error rate, which dictates the need to develop a hybrid model that combines the advantages of each model.

5.4. Open Research Issues

During the recent pandemic, adversaries relied more on unknown malwares in their attack techniques, where the utilisation of unknown attacks has increased from 20% before COVID-19 to 35% during the pandemic [44]. The current literature lacks a standard consistent definition of unknown attacks, where unseen instances are falsely treated as unknown attacks [10]. A new

categorisation method to define unknown attacks into two main categories (i.e., type-A and type-B) was proposed [10]. Given the ubiquity of IoT networks, the prevalence of botnet attacks and the importance of the research issue, in this study, the authors tested the previously proposed categorisation on detecting unknown botnet attacks in IoT networks.

Up to the authors' knowledge, this issue has not been addressed in the literature before; thus, it constitutes an open research issue. The results of this study indicated that current modern ML algorithms are not capable of detecting all type of unknown botnet attacks. It also highlighted the importance of proposing novel methods to solve the research issue. Although this study provides the analytical analysis of the research problem, a solution is yet to be proposed to fully address the problem of detecting unknown botnet attacks in IoT networks.

6. CONCLUSION AND FUTURE WORK

Unknown cybersecurity attacks remains a challenging issue yet to be solved by the research community. In this paper, the authors addressed the issue of detecting unknown botnet-based attacks that target IoT devices and empirically demonstrated that the current methods in detecting unknown attacks could not detect all types of actual unknown attacks. The performance of shallow and deep learning models in detecting unknown botnet attacks in IoT environment was evaluated.

A well-established and current benchmarking dataset (i.e., Aposemat IoT-23 dataset) was utilised in the evaluation. The research problem was formulated as both binary and multiclass supervised classification problem. The notion of type-A and type-B unknown attacks was applied in this research study, where type-A represents an entirely new unknown attack. In contrast, type-B represents an unknown subtype of attack of a previously known attack category.

The results showed that while some types of unknown botnet attacks were best discovered using shallow learners (e.g., "Linux.Mirai" C&C), others were detected the best using deep classifiers (e.g., Okiru). Moreover, while most botnet attacks were better detected using binary classifiers, other botnets, such as Kenjiro Okiru, were best discovered using multiclass models. Furthermore, the results indicated that no single classifier could correctly identify all types of unknown attacks with acceptable generalisation error metrics, where the overall weighted average classification error rate was considerably high (61.46–86.40%). This emphasises the need for a new architecture that combines the best of each model to detect unknown attacks with sufficient accuracy.

In future work, the authors plan to evaluate anomaly-based unsupervised deep learning classifiers in detecting unknown botnet attacks on IoT devices using the same methodology applied during this study. Moreover, to assess the performance of none -artificial neural networks-based machine learning algorithms in detected unknown attacks utilising the notion of type-A and type-B unknown attacks.

REFERENCES:

[1] M. D. Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "Analysis of DDoS-capable IoT malwares," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 807–816. doi: 10.15439/2017F288.

[2] P. Santikellur, T. Haque, M. Al-Zewairi, and R. S. Chakraborty, "Optimized Multi-Layer Hierarchical Network Intrusion Detection System with Genetic Algorithms," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, Oct. 2019, pp. 1–7. doi: 10.1109/ICTCS.2019.8923067.

[3] N. Moustafa, B. Turnbull, and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019, doi: 10.1109/JIOT.2018.2871719.

[4] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, Apr. 2017, doi: 10.1016/j.jnca.2017.02.009.

[5] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics*, vol. 9, no. 4, Art. no. 4, Apr. 2020, doi: 10.3390/electronics9040629.

[6] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," *Electronics*, vol. 9, no. 7, Art. no. 7, Jul. 2020, doi: 10.3390/electronics9071177.

[7] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture," *Sensors*, vol. 20, no. 16, Art. no. 16, Jan. 2020, doi: 10.3390/s20164372.

[8] Y. Meidan *et al.*, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.

[9] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, Feb. 2019, doi: 10.1016/j.jnca.2018.12.006.

[10] M. Al-Zewairi, S. Almajali, and M. Ayyash, "Unknown Security Attack Detection Using Shallow and Deep ANN Classifiers," *Electronics*, vol. 9, no. 12, Art. no. 12, Dec. 2020, doi: 10.3390/electronics9122006.

[11] Agustin Parmisano, Sebastian Garcia, and Maria Jose Erquiaga, "A labeled dataset with malicious and benign IoT network traffic," *Stratosphere Laboratory*, Jan. 2019, Accessed: Feb. 01, 2021. [Online]. Available: https://www.stratosphereips.org/datasets-iot23

[12] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommun Syst*, vol. 70, no. 3, pp. 447–489, Mar. 2019, doi: 10.1007/s11235-018-0475-8.

[13] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[14] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, p. 18, Mar. 2021, doi: 10.1186/s42400-021-00077-7.

[15] S. S. Gopalan, A. Raza, and W. Almobaideen, "IoT Security in Healthcare using AI: A Survey," in *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Mar. 2021, pp. 1–6. doi: 10.1109/ICCSPA49915.2021.9385711.

[16] A. J. Meera, M. V. V. P. Kantipudi, and R. Aluvalu, "Intrusion Detection System for the IoT: A Comprehensive Review," in *Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019)*, Dec. 2019, pp. 235–243. doi: 10.1007/978-3-030-49345-5_25.

[17] P. M. Chanal and M. S. Kakkasageri, "Security and Privacy in IoT: A Survey," *Wireless Pers Commun*, vol. 115, no. 2, pp. 1667–1693, Nov. 2020, doi: 10.1007/s11277-020-07649-9.

[18] S. Zaman, H. Tauqeer, W. Ahmad, S. M. A. Shah, and M. Ilyas, "Implementation of Intrusion Detection System in the Internet of Things: A Survey," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Nov. 2020, pp. 1–6. doi: 10.1109/INMIC50486.2020.9318047.

[19] V. Shakhov, O. Sokolova, and I. Koo, "A Criterion for IDS Deployment on IoT Edge Nodes," in *Computational Science and Its Applications – ICCSA 2020*, Jul. 2020, pp. 546–556. doi: 10.1007/978-3-030-58799-4_40.

[20] A. Uprety and D. B. Rawat, "Reinforcement Learning for IoT Security: A Comprehensive Survey," *IEEE Internet of Things Journal*, pp. 1–1, 2020, doi: 10.1109/JIOT.2020.3040957.

[21] V. Ponnusamy, S. Bakhshad, B. Sharma, R. Annur, and T. B. Seong, "A Survey on Intrusion Detection in Wired and Wireless Network for Future IoT Deployment," pp. 119--144, 2020, doi: 10.4018/978-1-7998-2803-7.ch007.

[22] M. Behniafar, A. R. Nowroozi, and H. R. Shahriari, "A Survey of Anomaly Detection Approaches in Internet of Things," *The ISC International Journal of Information Security*, vol. 10, no. 2, pp. 79–92, Jul. 2018, doi: 10.22042/isecure.2018.116976.408.

[23] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Implementing Lightweight IoT-IDS on Raspberry Pi Using Correlation-Based Feature Selection and Its Performance Evaluation," in *Advanced Information Networking and Applications*, Mar. 2019, pp. 458–469. doi: 10.1007/978-3-030-15032-7_39.

[24] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "A Sequential Scheme for Detecting Cyber Attacks in IoT Environment," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*,

Aug. 2019, pp. 238–244. doi: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00051.

[25] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things," *Sensors*, vol. 20, no. 2, Art. no. 2, Jan. 2020, doi: 10.3390/s20020461.

[26] I. Almomani and A. Alromi, "Integrating Software Engineering Processes in the Development of Efficient Intrusion Detection Systems in Wireless Sensor Networks," *Sensors*, vol. 20, no. 5, Art. no. 5, Jan. 2020, doi: 10.3390/s20051375.

[27] X. C. Yin, Z. G. Liu, L. Nkenyereye, and B. Ndibanje, "Toward an Applied Cyber Security Solution in IoT-Based Smart Grids: An Intrusion Detection System Approach," *Sensors*, vol. 19, no. 22, Art. no. 22, Jan. 2019, doi: 10.3390/s19224952.

[28] G. Vaidya, A. Nambi, T. V. Prabhakar, V. K. T, and S. Sudhakara, "IoT-ID: A Novel Device-Specific Identifier Based on Unique Hardware Fingerprints," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2020, pp. 189–202. doi: 10.1109/IoTDI49375.2020.00026.

[29] A. Alhowaide, I. Alsmadi, and J. Tang, "PCA, Random-Forest and Pearson Correlation for Dimensionality Reduction in IoT IDS," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Sep. 2020, pp. 1–6. doi: 10.1109/IEMTRONICS51293.2020.9216388.

[30] Y. Otoum and A. Nayak, "AS-IDS: Anomaly and Signature Based IDS for the Internet of Things," *J Netw Syst Manage*, vol. 29, no. 3, p. 23, Mar. 2021, doi: 10.1007/s10922-021-09589-6.

[31] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The Use of Ensemble Models for Multiple Class and Binary Class Classification for Improving Intrusion Detection Systems," *Sensors*, vol. 20, no. 9, Art. no. 9, Jan. 2020, doi: 10.3390/s20092559.

[32] R. Qaddoura, A. M. Al-Zoubi, I. Almomani, and H. Faris, "A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling," *Applied Sciences*, vol. 11, no. 7, Art. no. 7, Jan. 2021, doi: 10.3390/app11073022.

[33] B. D, KiblerDennis, P. J, and SmythPadhraic, "The UCI KDD archive of large data sets for data mining research and experimentation," *ACM SIGKDD Explorations Newsletter*, Dec. 2000, doi: 10.1145/380995.381030.

[34] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[35] B. Sangster *et al.*, "Toward instrumenting network warfare competitions to generate labeled datasets," in *Proceedings of the 2nd conference on Cyber security experimentation and test*, USA, Aug. 2009, p. 9.

[36] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Apr. 2021, pp. 108–116. Accessed: Apr. 13, 2021. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0006639801080116

[37] W. Haider, J. Hu, J. Slay, B. P. Turnbull, and Y. Xie, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling," *Journal of Network and Computer Applications*, vol. 87, pp. 185–192, Jun. 2017, doi: 10.1016/j.jnca.2017.03.018.

[38] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.

[39] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[40] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.

[41] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental Evaluation of a Multi-layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Oct. 2017, pp. 167–172. doi: 10.1109/ICTCS.2017.29.

[42] K. Pasupa and W. Sunhem, "A comparison between shallow and deep architecture classifiers on small dataset," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct. 2016, pp. 1–6. doi: 10.1109/ICITEED.2016.7863293.

[43] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey," *arXiv:1701.02145 [cs]*, Jan. 2017, Accessed: Apr. 16, 2021. [Online]. Available: http://arxiv.org/abs/1701.02145

[44] C. Nabe, "Impact of COVID-19 on Cybersecurity," Deloitte. Accessed: Jun. 06, 2021. [Online]. Available: https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

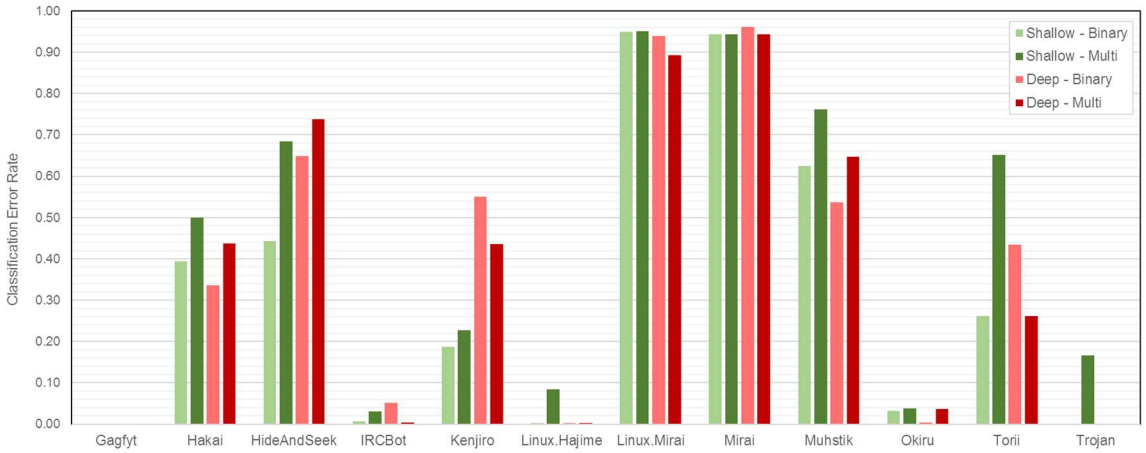


Figure 6: Comparison between the classification error rate for type-A unknown attacks.

Table 6: Evaluation metrics for binary shallow and deep DL models in detecting type-A unknown attacks.

Label		Shallow				Deep			
Class	**Instances**	**Recall**	**Accuracy**	**F1**	**Error**	**Recall**	**Accuracy**	**F1**	**Error**
Gagfyt	65,619	0.999	0.999	0.999	0.001	0.999	0.999	0.999	0.001
Hakai	4112	0.606	0.606	0.755	0.394	0.665	0.665	0.799	0.335
HideAndSeek	266,956	0.556	0.556	0.715	0.444	0.352	0.352	0.521	0.648
IRCBot	30,343	0.993	0.993	0.996	0.007	0.948	0.948	0.973	0.052
Kenjiro	18,532	0.813	0.813	0.897	0.187	0.450	0.450	0.621	0.550
Linux.Hajime	131,477	0.998	0.998	0.999	0.002	0.998	0.998	0.999	0.002
Linux.Mirai	56,792	0.051	0.051	0.097	0.949	0.062	0.062	0.116	0.938
Mirai	3,919,807	0.057	0.057	0.107	0.943	0.038	0.038	0.074	0.962
Muhstik	113,959	0.375	0.375	0.545	0.625	0.463	0.463	0.633	0.537
Okiru	16,311	0.968	0.968	0.984	0.032	0.996	0.996	0.998	0.004
Torii	23	0.739	0.739	0.850	0.261	0.565	0.565	0.722	0.435
Trojan	6	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000

Table 7: Evaluation metrics for multiclass shallow and deep DL models in detecting type-A unknown attacks.

Label		Shallow				Deep			
Class	**Instances**	**Recall**	**Accuracy**	**F1**	**Error**	**Recall**	**Accuracy**	**F1**	**Error**
Gagfyt	65,619	0.999	0.999	1.000	0.001	0.999	0.999	0.999	0.001
Hakai	4112	0.500	0.500	0.667	0.500	0.562	0.562	0.720	0.438
HideAndSeek	266,956	0.316	0.316	0.480	0.684	0.262	0.262	0.416	0.738
IRCBot	30,343	0.969	0.969	0.984	0.031	0.996	0.996	0.998	0.004
Kenjiro	18,532	0.773	0.773	0.872	0.227	0.565	0.565	0.722	0.435
Linux.Hajime	131,477	0.916	0.916	0.956	0.084	0.998	0.998	0.999	0.002
Linux.Mirai	56,792	0.049	0.049	0.093	0.951	0.107	0.107	0.193	0.893
Mirai	3,919,807	0.057	0.057	0.107	0.943	0.056	0.056	0.107	0.944
Muhstik	113,959	0.238	0.238	0.385	0.762	0.353	0.353	0.522	0.647
Okiru	16,311	0.962	0.962	0.981	0.038	0.964	0.964	0.982	0.036
Torii	23	0.348	0.348	0.516	0.652	0.739	0.739	0.850	0.261
Trojan	6	0.833	0.833	0.909	0.167	1.000	1.000	1.000	0.000

Table 8: Evaluation metrics for binary shallow and deep DL models in detecting type-B unknown attacks.

Label			Shallow				Deep			
Class	**Subclass**	**Instances**	**Recall**	**Accuracy**	**F1**	**Error**	**Recall**	**Accuracy**	**F1**	**Error**
Gagfyt	CC-HeartBeat	95	0.453	0.453	0.623	0.547	0.537	0.537	0.699	0.463
	DDoS	65,524	0.439	0.439	0.610	0.561	0.155	0.155	0.269	0.845
Hakai	CC	4112	0.513	0.513	0.678	0.487	0.560	0.560	0.718	0.440
HideAndSeek	CC	1	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
	PartOfAHorizontalPortScan	266,955	0.312	0.312	0.476	0.688	0.296	0.296	0.456	0.704
IRCBot	Attack	677	0.957	0.957	0.978	0.043	0.786	0.786	0.880	0.214
	CC	1423	0.706	0.706	0.828	0.294	0.871	0.871	0.931	0.129
	PartOfAHorizontalPortScan	28,243	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
Kenjiro	Attack	4	0.000	0.000	0.000	1.000	0.250	0.250	0.400	0.750
	CC-HeartBeat	9947	0.841	0.841	0.913	0.159	0.674	0.674	0.805	0.326
	DDoS	1094	0.402	0.402	0.574	0.598	0.995	0.995	0.998	0.005
	Okiru	3004	0.027	0.027	0.052	0.973	0.140	0.140	0.246	0.860
	PartOfAHorizontalPortScan	4478	0.874	0.874	0.933	0.126	0.299	0.299	0.461	0.701
	PartOfAHorizontalPortScan-Attack	5	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
Linux.Hajime	PartOfAHorizontalPortScan	131,477	0.998	0.998	0.999	0.002	0.998	0.998	0.999	0.002
Linux.Mirai	CC-HeartBeat	5777	0.997	0.997	0.998	0.003	0.448	0.448	0.618	0.552
	DDoS	39,584	0.001	0.001	0.003	0.999	0.000	0.000	0.000	1.000
	Okiru	11,431	0.993	0.993	0.996	0.007	0.970	0.970	0.985	0.030
Mirai	Attack	2755	0.448	0.448	0.619	0.552	0.649	0.649	0.787	0.351
	CC	12,093	0.748	0.748	0.856	0.252	0.575	0.575	0.730	0.425
	CC-FileDownload	50	0.960	0.960	0.980	0.040	0.960	0.960	0.980	0.040
	CC-HeartBeat-Attack	834	0.474	0.474	0.643	0.526	0.312	0.312	0.475	0.688
	CC-HeartBeat-FileDownload	11	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
	CC-Mirai	2	0.000	0.000	0.000	1.000	0.500	0.500	0.667	0.500

	CC-PartOfAHorizontalPortScan	795	1.000	1.000	1.000	0.000	0.605	0.605	0.754	0.395
	DDoS	2,076,658	0.032	0.032	0.061	0.968	0.063	0.063	0.119	0.937
	FileDownload	15	0.867	0.867	0.929	0.133	0.000	0.000	0.000	1.000
	Okiru	227	0.295	0.295	0.456	0.705	0.203	0.203	0.337	0.797
	PartOfAHorizontalPortScan	1,826,367	0.084	0.084	0.154	0.916	0.079	0.079	0.147	0.921
Muhstik	Attack	5962	0.065	0.065	0.123	0.935	0.955	0.955	0.977	0.045
	CC	8	0.375	0.375	0.545	0.625	0.250	0.250	0.400	0.750
	PartOfAHorizontalPortScan	107,989	0.249	0.249	0.398	0.751	0.435	0.435	0.607	0.565
Okiru	CC-HeartBeat	15,687	0.923	0.923	0.960	0.077	1.000	1.000	1.000	0.000
	Okiru	621	0.008	0.008	0.016	0.992	0.895	0.895	0.945	0.105
	Okiru-Attack	3	0.667	0.667	0.800	0.333	0.667	0.667	0.800	0.333
Torii	CC-Torii	23	0.739	0.739	0.850	0.261	0.348	0.348	0.516	0.652
Trojan	CC-FileDownload	3	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
	FileDownload	3	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000

Table 9: Evaluation metrics for multiclass shallow and deep DL models in detecting type-B unknown attacks.

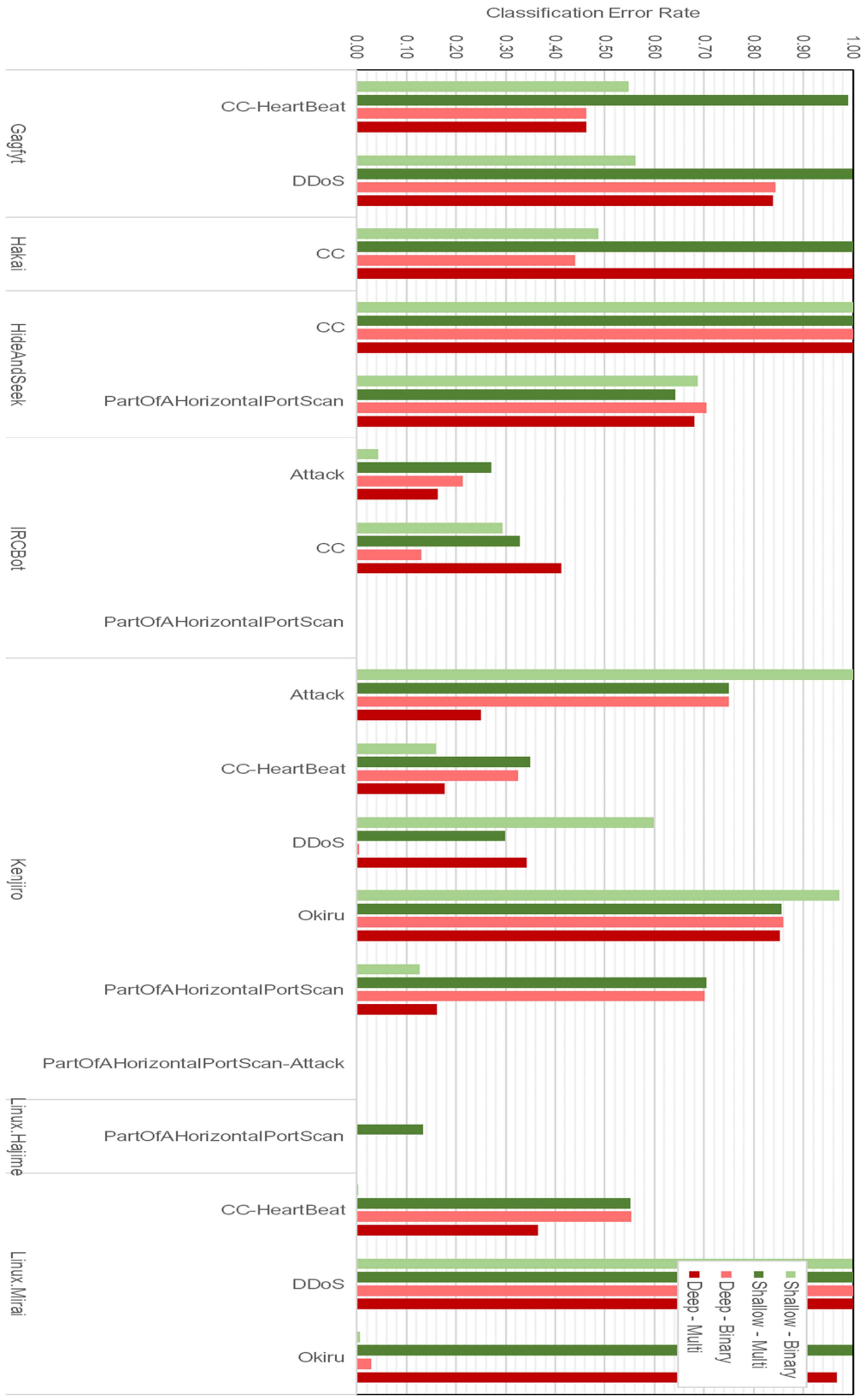
Label			Shallow				Deep			
Class	**Subclass**	**Instances**	**Recall**	**Accuracy**	**F1**	**Error**	**Recall**	**Accuracy**	**F1**	**Error**
Gagfyt	CC-HeartBeat	95	0.537	0.537	0.699	0.463	0.011	0.011	0.021	0.989
	DDoS	65,524	0.161	0.161	0.277	0.839	0.000	0.000	0.001	1.000
Hakai	CC	4112	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
HideAndSeek	CC	1	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
	PartOfAHorizontalPortScan	266,955	0.319	0.319	0.484	0.681	0.357	0.357	0.526	0.643
IRCBot	Attack	677	0.836	0.836	0.911	0.164	0.728	0.728	0.843	0.272
	CC	1423	0.588	0.588	0.741	0.412	0.672	0.672	0.804	0.328
	PartOfAHorizontalPortScan	28,243	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
Kenjiro	Attack	4	0.750	0.750	0.857	0.250	0.250	0.250	0.400	0.750
	CC-HeartBeat	9947	0.823	0.823	0.903	0.177	0.651	0.651	0.789	0.349
	DDoS	1094	0.658	0.658	0.794	0.342	0.700	0.700	0.824	0.300
	Okiru	3004	0.147	0.147	0.257	0.853	0.144	0.144	0.252	0.856

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	PartOfAHorizontalPortScan	4478	0.838	0.838	0.912	0.162	0.296	0.296	0.457	0.704
	PartOfAHorizontalPortScan-Attack	5	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
Linux.Hajime	PartOfAHorizontalPortScan	131,477	0.998	0.998	0.999	0.002	0.867	0.867	0.929	0.133
Linux.Mirai	CC-HeartBeat	5777	0.635	0.635	0.777	0.365	0.448	0.448	0.619	0.552
	DDoS	39,584	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
	Okiru	11,431	0.033	0.033	0.064	0.967	0.001	0.001	0.003	0.999
Mirai	Attack	2755	0.173	0.173	0.295	0.827	0.054	0.054	0.103	0.946
	CC	12,093	0.712	0.712	0.832	0.288	0.612	0.612	0.759	0.388
	CC-FileDownload	50	0.760	0.760	0.864	0.240	0.920	0.920	0.958	0.080
	CC-HeartBeat-Attack	834	0.547	0.547	0.707	0.453	0.034	0.034	0.065	0.966
	CC-HeartBeat-FileDownload	11	1.000	1.000	1.000	0.000	1.000	1.000	1.000	0.000
	CC-Mirai	2	0.000	0.000	0.000	1.000	0.500	0.500	0.667	0.500
	CC-PartOfAHorizontalPortScan	795	1.000	1.000	1.000	0.000	0.605	0.605	0.754	0.395
	DDoS	2,076,658	0.032	0.032	0.061	0.968	0.032	0.032	0.061	0.968
	FileDownload	15	0.867	0.867	0.929	0.133	0.733	0.733	0.846	0.267
	Okiru	227	0.216	0.216	0.355	0.784	0.291	0.291	0.451	0.709
	PartOfAHorizontalPortScan	1,826,367	0.079	0.079	0.146	0.921	0.079	0.079	0.146	0.921
Muhstik	Attack	5962	0.048	0.048	0.092	0.952	0.236	0.236	0.382	0.764
	CC	8	0.250	0.250	0.400	0.750	0.250	0.250	0.400	0.750
	PartOfAHorizontalPortScan	107,989	0.233	0.233	0.378	0.767	0.233	0.233	0.378	0.767
Okiru	CC-HeartBeat	15,687	0.093	0.093	0.171	0.907	0.218	0.218	0.358	0.782
	Okiru	621	0.048	0.048	0.092	0.952	0.897	0.897	0.946	0.103
	Okiru-Attack	3	0.333	0.333	0.500	0.667	0.333	0.333	0.500	0.667
Torii	CC-Torii	23	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
Trojan	CC-FileDownload	3	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
	FileDownload	3	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000

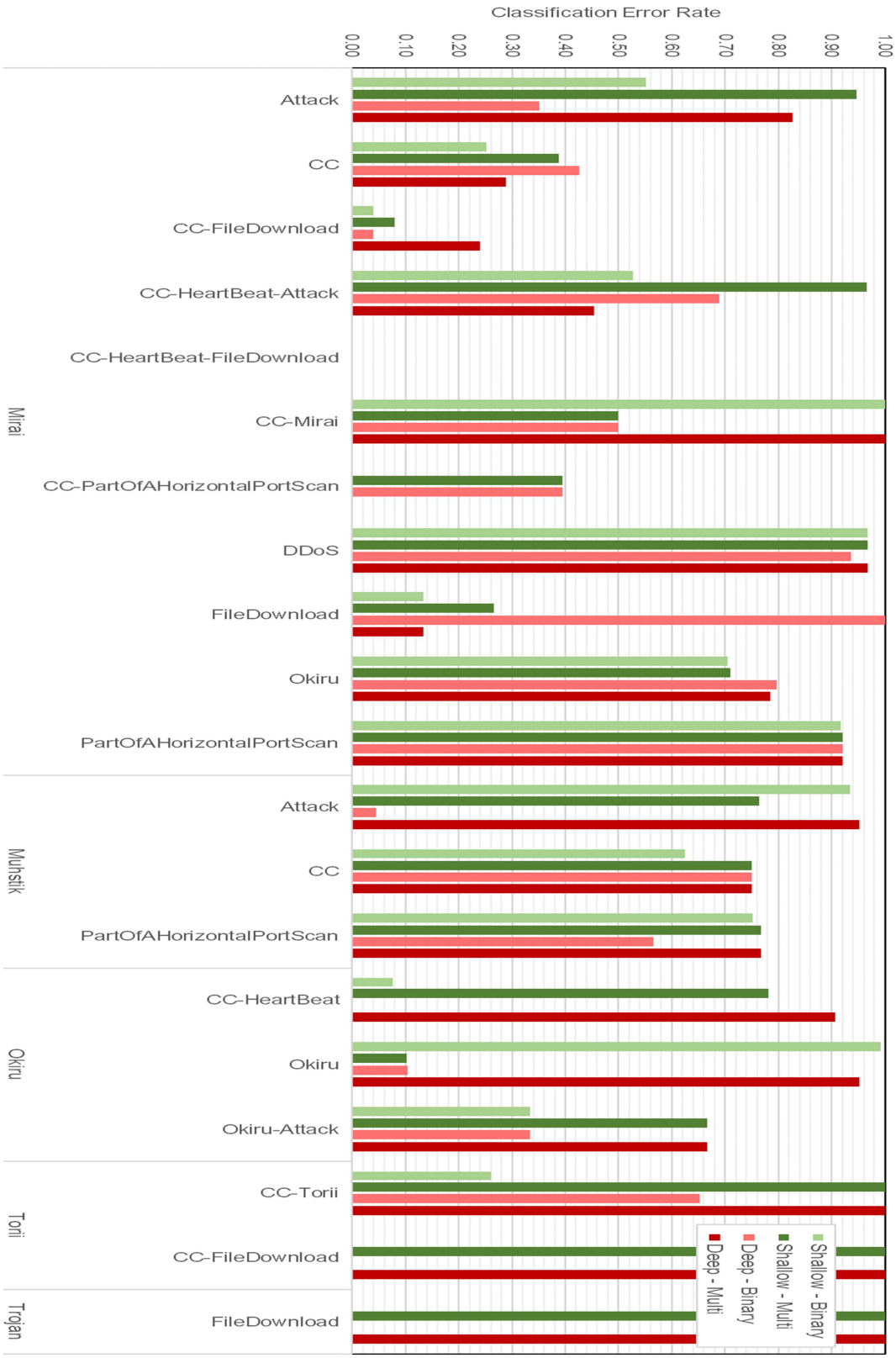
Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a)

Figure 7: Comparison between the classification error rate for type-B unknown attacks.



(b)

Figure 7: Comparison between the classification error rate for type-B unknown attacks.