Generating a benchmark cyber multi-step attacks dataset for intrusion detection

Article *in* Journal of Intelligent and Fuzzy Systems · April 2022

DOI: 10.3233/JIFS-213247

3 authors:

Mohammad Almseidin
University of Miskolc
32 PUBLICATIONS **618** CITATIONS
SEE PROFILE

Jamil Al Sawwa
Tafila Technical University
11 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Mouhammd Alkasassbeh
Princess Sumaya University for Technology
98 PUBLICATIONS **1,288** CITATIONS
SEE PROFILE

Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-213247
IOS Press

Generating a benchmark cyber multi-step attacks dataset for intrusion detection

Mohammad Almseidin[a,*], Jamil Al-Sawwa[b] and Mouhammd Alkasassbeh[c]
[a]*Computer Science Department, Aqaba University of Technology, Aqaba, Jordan*
[b]*Computer Science Department, Tafila Technical University, Tafila, Jordan*
[c]*Computer Science Department, Princess Sumaya University for Technology, Amman, Jordan*

Abstract. Nowadays, with the rapid increase in the number of applications and networks, the number of cyber multi-step attacks has been increasing exponentially. Thus, the need for a reliable and acceptable Intrusion Detection System (IDS) solution is becoming urgent to protect the networks and devices. However, implementing a robust IDS needs a reliable and up-to-date dataset in order to capture the behaviors of the new types of attacks especially a multi-step attack. In this paper, a new benchmark Multi-Step Cyber-Attack Dataset (MSCAD) is introduced. MSCAD includes two multi-step scenarios; the first scenario is a password cracking attack, and the second attack scenario is a volume-based Distributed Denial of Service (DDoS) attack. The MSCAD was assessed in two manners; firstly, the MSCAD was used to train IDS. Then, the performance of IDS was evaluated in terms of G-mean and Area Under Curve (AUC). Secondly, the MSCAD was compared with other free open-source and public datasets based on the latest keys criteria of a dataset evaluation framework. The results show that IDS-based MSCAD achieved the best performance with G-mean 0.83 and obtained good accuracy to detect the attacks. Besides, the MSCAD successfully passing twelve keys criteria.

Keywords: Intrusion detection system (IDS), multi-step cyber-attacks, machine learning, resampling algorithms, intrusion datasets

1. Introduction

The multi-step attack [1–4] is an attack implemented based on several prerequisite steps to achieve the final goal. Typically, the attacker performs this type of attack to avoid any detection because some prerequisite steps are considered somehow a normal behavior. As reported in the Chinese network security report in 2018 [1, 5, 6], the multi-step attacks represent 60% of the total number of attacks around the world.

In the last decade, computers and network security devices suffer from genuine attacks. The Distributed Denial of Service (DDoS) attack is one of the most serious multi-step attacks, which aims to target the victim's network (or resources) with malicious traffic that causes the network to be overwhelmed [7–9]. In May 2017, the intruders implement a new attack which is called the Wannacry ransomware attack [1]. A large proportion of intrusion detection systems failed to detect this attack because it acts as a legitimate user request in an intelligent way. Specifically, this attack has several normality steps to launch itself. To secure computers and network devices from the attacks, the attackers' behaviors and techniques should be studied and investigated as well as the newest detection approaches should be evaluated with a realistic and up-to-date dataset.

In general, IDS techniques could be categorized as either monitoring-based or detection-based. In the monitoring-based technique, two systems are used, which are Network Intrusion Detection System

*Corresponding author. Mohammad Almseidin, Computer Science Department, Aqaba University of Technology, Aqaba, Jordan. E-mail: msoudi@aut.edu.jo.

(NIDS) and Host Intrusion Detection System (HIDS). NIDS operates to monitor and detect intrusions within all network devices and protect the whole network traffic while HIDS is implemented to protect a specific device [10]. According to the detection-based technique, there are two types of IDS; anomaly-based and signature-based [11]. The anomaly-based IDS is implemented to detect intrusions based on the records of the historical normal behavior of a specific network. Consequently, anomaly-based IDS compares the real-time traffic with the recorded behavior of traffic to detect intrusions. In the signature-based IDS, IDS verifies each real-time packet based on the intrusions' signatures repository . Therefore, it is needed to be frequently updated with the latest intrusion signatures in order to detect the new intrusion signatures, otherwise, the new intrusion signature could be passed through IDS successfully. Designing and implementing a reliable IDS is becoming a persistent demand to handle various attack scenarios especially multi-step attacks. Nevertheless, the IDS needs an up-to-date dataset to built and train the IDS [9, 12, 13].

This paper introduces a new benchmark multi-step cyber-attack dataset (MSCAD). The introduced MSCAD was evaluated in two manners; firstly, MSCAD was used with the state-of-art machine learning algorithms; Decision Tree and Random Forest, to built and train an IDS. Then, the performance of IDS for detecting the attacks was assessed in terms of G-mean and AUC. Secondly, the MSCAD is compared with other free open-source and publicly available datasets based on the latest keys criteria of the dataset evaluation framework. The experimental results show that the MSCAD successfully passing twelve keys criteria, which demonstrates its effectiveness in reflecting the real behavior of multi-step attack scenarios. Besides, MSCAD could be very useful for building a reliable and acceptable IDS for detecting multi-step attacks, and it would be freely available to the researcher's community.

The rest of the paper is organized as follows: section 2 presents the recent works related to developing a benchmark dataset for intrusion detection. In addition, it provides a technical discussion about the shortcomings of the current intrusion datasets and how the MSCAD handle these shortcomings. Section 3 details the design and generation of the proposed multi-step cyber-attack dataset. The results and discussion presented in section 4. Finally, section 6 concludes the paper.

2. IDS benchmark intrusion datasets

This section presents some of the recent relevant works related to developing a benchmark dataset for intrusion detection. Furthermore, it provides a technical discussion about the shortcomings of the current intrusion datasets and how the MSCAD handles these shortcomings.

Typically, the intrusion detection datasets are collected either in packet-based or flow-based format. In the packet-based format, the data includes complete payload information based on the transport protocols. This format is usually captured as packet capture (PCAP) files. Oppositely, the data in the flow-based format includes a meta-information about network connections within a time window as well as five identifier attributes; source IP address, source port, destination IP address, destination port, and transport protocol [4, 10, 13, 14].

The intrusion detection datasets provide a convenient environment for the research community to evaluate their techniques and models for intrusion detection. However, assessing the suggested detection approaches with out-of-date datasets could not reflect the actual performance of these approaches for detecting the recent types of attacks, and it could lead to unrealistic results. The recent works that are relevant to generate benchmark datasets for intrusion detection are summarized as follows [14–16]:

- **KDD-98**
  In 1998, the first public intrusion detection dataset was made by Defence Advanced Research Project Agency (DARPA), which called Knowledge Discovery and Data Mining (KDD-98) [17]. The KDD-98 dataset contains 41 features including the fundamental features of TCP connections and other features such as the number of failed logins, wrong fragment, and host login. In addition, it contains around 5 million records that are categorized into four groups of attacks; Denial of Service (DoS), Remote to Local (R2L), the user to root, and probing attacks. Until 2018, 63.8% of the research community used this dataset to evaluate their proposed detection approaches [16]. However, KDD-98 suffers from significant drawbacks that affect the performance of proposed detection approaches such as duplicated records. Besides, the types of attacks covered in this dataset are considered out-of-

date and do not reflects the current security issues in network traffic [18–20].

- **NSL-KDD**
  This dataset is an enhanced version of the KDD-98 dataset and was introduced in 2009 by Tavallaee et al. [18]. The aim behind this enhancement is to remove the redundant records and presented a more realistic dataset. NSL-KDD dataset includes only 150,000 records after removing the redundant records, which could be downloaded freely from [19]. However, the enhanced version of the KDD-99 dataset still has the drawbacks such as out-of-date types of attacks (since 1998) and the network parameters (features) do not reflect the real network traffic. Nevertheless, NSL-KDD still used as a benchmark dataset within the IDS research community.
- **CAIDA** CAIDA dataset was introduced in 2007 by Hick et al. [21]. This dataset available as network trace files includes only DDoS attacks. In addition, the CAIDA dataset does not include the network parameters (features) that could lead to a lack of knowledge about the real behavior of DDoS attacks and make the detection of abnormal traffic is difficult.
- **UNIBS** UNIBS dataset was introduced in 2009 by Gringoli et al. [22]. This dataset is mainly concerned to define the main characteristic of the behavior of web applications such as Skype and web browsers. UNIBS dataset contains around 79,000 flow-based records without intrusions' behaviors. Besides, this dataset could not be used for identifying the abnormality within computer networks. Therefore, this dataset is still used as background traffic within the IDS simulated environment for security purposes.
- **ISCX 2012** ISCX 2012 dataset was generated within a simulated network environment in 2012 by Shiravi et al. in [19]. It includes two network profiles; α and β profile. α profile defines the attacker scenarios while β profile defines the normal activities. In ISCX 2012 dataset, a dynamic approach was used to introduce various attack scenarios such as SSH brute force and DoS. However, ISCX 2012 dataset does not include any HTTPS traffic, although HTTPS protocol represents 70% of network traffic.
  DDoS 2016 dataset was introduced originally in 2016 by Alkassasbeh et al. [23]. This dataset was generated using Network Simulator (NS2). It includes different types of DDoS attacks such as UDP flood and SIDDOS attacks. DDoS 2016 dataset could be categorized as flow-based format, which includes around two million records with 21 network parameters. DDoS 2016 dataset concerned with DDoS attacks which cause the services to be interrupted for the end-users. The major drawback of the DDoS 2016 dataset is that is not suitable for detecting multi-step attacks because it does not include any sequence attack steps. Moreover, DDoS 2016 dataset includes somehow out-of-date attacks with a large number of statistical features. These statistical features could not be extracted directly from the network traffic, but it is calculated separately, which could lead to the time and resource-consuming.
- **ADFA-LD and ADFA-WD** Two datasets; ADFA Linux Dataset (ADFA-LD) and ADFA Windows Dataset (ADFA-WD) were created by the Australian Defence Force Academy (ADFA) using the system call traces files [24]. ADFA-LD dataset includes zero-day malware records. ADFA-WD is mainly used to evaluate the HIDS and includes a number of network parameters such as source bits per second, destination bits per second, and destination packet count. More details about the features and system call traces could be found in [24]. However, the ADFA-LD and ADFA-WD datasets are generated based on a large number of calculated statistical features. Furthermore, many attack techniques can bypass system trace files by behaving as legitimate behavior to avoid any detection.
- **CIC-IDS 2017** CIC-IDS 2017 dataset was created over five days using a simulated test-bed environment and includes 80 network parameters [25]. This dataset investigated the behavior of various attacks; Web Attack, Infiltration attack, DoS, and Heart-bleed. The labeling process for network traffic was accomplished based on timestamp, protocols, and the source and destination ports. However, the CIC-IDS 2017 dataset does not have any behavior of sequence steps of the multi-step attack scenarios. Moreover, according to [26], the CIC-IDS 2017 dataset includes a large number of missing values (288602 records) that could lead to unrealistic results and does not reflect the real network traffic.

– **TUIDS** TUIDS dataset was initially generated by the Tezpur University based on the mixed of packet-based and bidirectional flow-based format. TUIDS dataset includes three types of network traffic; the TUIDS intrusion dataset, TUIDS scan dataset, and TUIDS DDoS dataset. The simulated environment contains 250 hosts in order to generate the port scan and DDoS attacks [27]. The major drawback of the TUIDS dataset is that not simulating the sequence steps of the multi-step attack. Furthermore, according to [28, 29], the TUIDS dataset does not have the identifiers features such as source address, a destination address, source ports, and destination ports.

2.1. Discussion

The previous research works provided plausible contributions and supported the idea that generating an up-to-date multi-step attacks dataset is necessary for detecting and forecasting the intrusion at the early stage. However, the previously proposed intrusion datasets suffer from the shortcomings that summarized as follows:

– The studied intrusion datasets [17, 18, 21–23, 25, 27] do not have the sequence steps of attack scenarios. Thus, these datasets are not suitable for building IDS to detect multi-step attacks.
– In [17, 19], the proposed intrusion datasets contain a large number of redundant records and missing values. Thus, this could lead to unrealistic results and do not reflect real-world network traffic.
– The studied intrusion datasets [17, 18, 21–23, 25, 27] do not include the recent intrusions' techniques such as Web Crawling, Volume-based DDoS, and App-based DDoS. In addition, the intrusion dataset in [19] does not include an HTTPS network protocol.
– Some intrusion datasets such as [21, 22] concerned only with a specific type of attack and discard other attack types. Thus, the IDS systems based on these datasets are designed for a specific attack type without considering the other attack types. Furthermore, the acceptable IDS system should deal with the different behavior of attacks, especially the multi-step attack.
– The intrusion datasets in [23, 27] include a large number of statistical features that could lead to time- and resource-consuming. Furthermore, the intrusion dataset in [21] does not have any network parameter (feature). So, detecting abnormal traffic becomes difficult.

In response to the previous shortcomings, this paper introduces a new benchmark Multi-Step Cyber Attack (MSCAD) dataset for intrusion detection systems. The MSCAD overcomes the previously mentioned shortcomings as follows:

– The MSCAD is suitable for detecting multi-step attacks because of having various attack scenarios. These scenarios were launched based on sequential attacker steps. This technique of attack scenario generation provides a suitable environment for intrusion detection forecasting based on different algorithms for improving cyber defense.
– The MSCAD is homogeneous, which includes the recent intrusions' techniques; Web Crawling, Volume-based DDoS, and App-based DDoS.
– The MSCAD includes various network protocols including HTTPS protocol. Besides, it is generated from the complete network configuration.
– The MSCAD is available in two formats (PCAPs and CSV's). The PCAP files are available to be processed for IDS offline mode (replay mode). Moreover, The MSCAD is completely labeled together with 77 network parameters (features) that make the detection of abnormal traffic is applicable as well as presenting the features that could be affected directly by the studied attack scenarios.
– The MSCAD does not have redundant records and missing values. Therefore, this dataset does not need to be preprocessed before training the IDS system.

3. Multi-step cyber-attacks dataset

This section introduces the multi-step cyber attacks dataset generation in detail besides describing the tzest-bed environment and the studied attack scenarios.

3.1. Test-bed network architecture

The test-bed environment of the MSCAD dataset consists of two networks; the attacker-network and

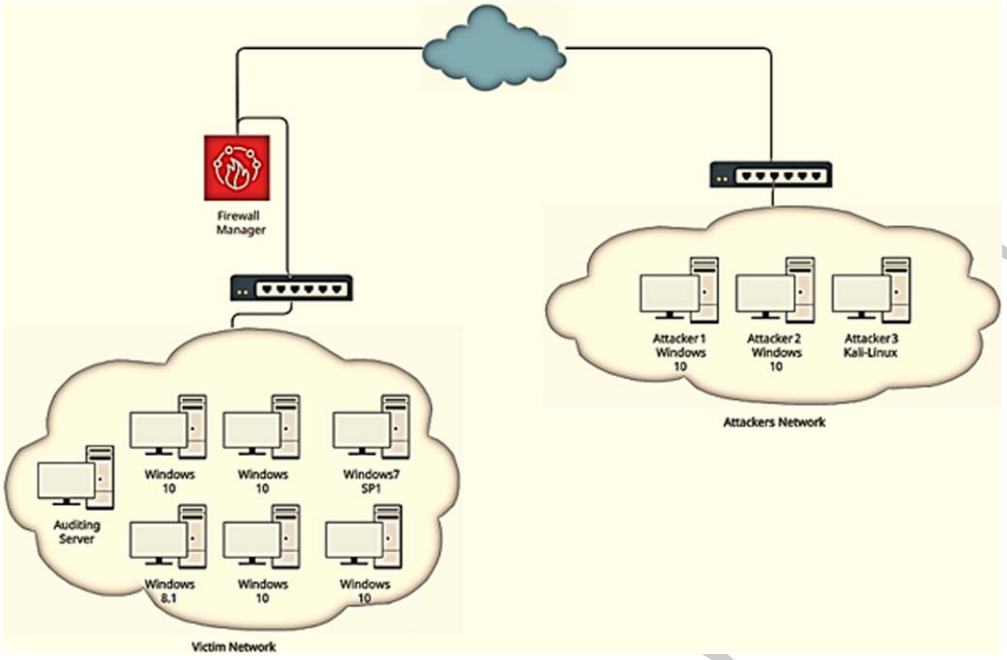


Fig. 1. The Test-bed environment architecture.

the victim-network, as shown in Fig. 1. From this figure, we can see that the two networks are completely separated. The victim-network consists of a collection of PCs and network devices (switches, routers, etc) that are secured by Firewall. While the attacker-network infrastructure consists of a collection of PCs in order to execute an attack on any host within the victim-network. The aim behind of this separation is to reflect the real cyber attack scenario when the victim-network is protected by the security solutions such as Firewall. The implemented test-bed environment has different tools and operating systems to reflect the real factors of network security. The detailed information about the victim-network operating system specifications and the attacker-network running scripts are shown in the Tables 1 and 2 respectively.

3.2. Multi-step attacks scenarios

The MSCAD includes two multi-step cyber-attacks scenarios as shown in Fig. 2. The two multi-step attack scenarios were performed as follows:

- **Multi-step Attack Scenario A**: In this scenario, an attacker aims to perform a password cracking attack (Brute force) on any host within the victim-network. The attacker executes this attack in three main sequential steps as presented in Table 3. Firstly, the port scan was executed simultaneously as shown in Table 4.

  Secondly, the HTTrack Website Copier[1] was used as a website crawler tool to take offline copy of the web application pages. By using a password list of 47 entries and user list of 10 entries that resulted in 470 attempts to crack the password. Finally, the Brute force script was executed as follows:

```
nmap -T5 --script http-form-brute
--script-args http-form-brute.
path=/bWAPP/login.php,http-
form-brute.method=POST,http-form
-brute.uservar=login -p 80
[IP-ADDRESS]
```

  It is worth mentioning that this scenario took three hours to be executed. Three hosts in the victim network were infected by this attack, which are 192.168.159.131, 192.168.159.12, and 192.168.159.14 .
- **Multi-step Attack Scenario B**: In scenario B, the attacker aims to execute the volume-based DDoS on any host within the victim network.

[1] https://www.httrack.com/

Table 1
Victim network operating systems specification

Client	Operating System	Description	IP Address
1	Windows 10	Victim Network	192.168.159.131
2	Windows 10	Victim Network	192.168.159.14
3	Windows 10	Victim Network	192.168.159.10
4	Windows 10	Victim Network	192.168.159.11
5	Windows 7 SP1	Victim Network	192.168.159.12
6	Windows 8.1	Victim Network	192.168.159.13
7	Windows 10 (Auditing)	Victim Network	192.168.159.14

Table 2
The attacker's network running scripts

Attacker	Operating System	Description	Running Scripts
1	Windows 10	Attacker Network	Port Scan, ICMP Flood, HTTP DDoS, Brute Force, Web Crawler
2	Windows 10	Attacker Network	Port Scan, ICMP Flood, HTTP DDoS, Brute Force, Web Crawler
3	Kali Linux	Attacker Network	Port Scan, ICMP Flood, HTTP DDoS, Brute Force, Web Crawler

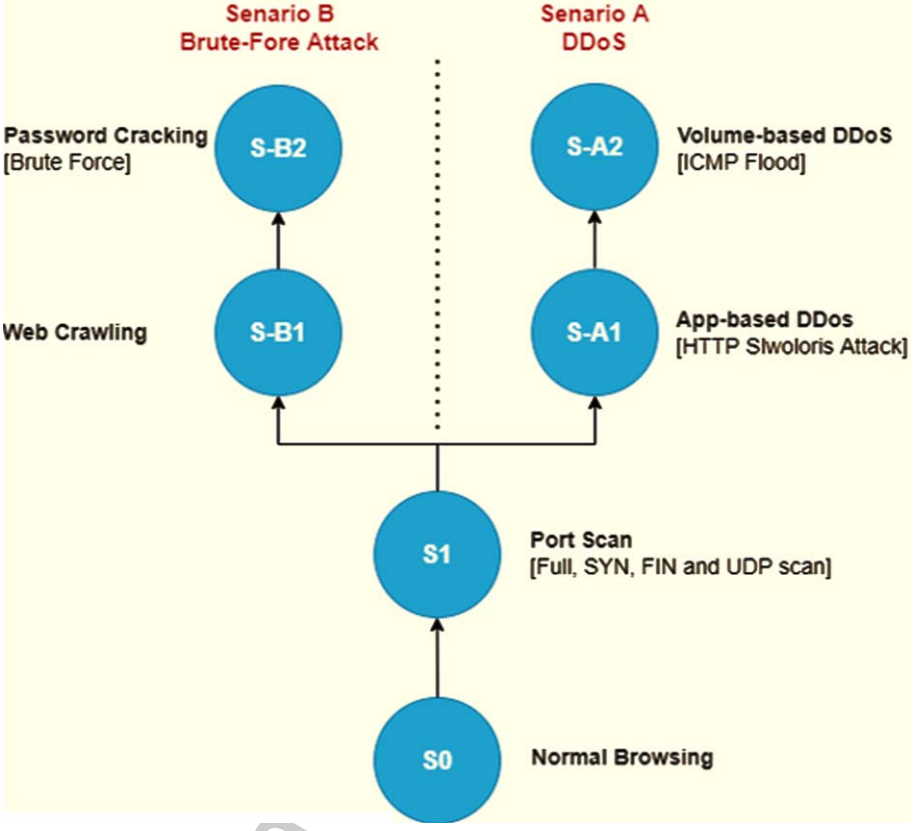


Fig. 2. The sequence steps of the studied attacks scenarios.

Table 3
The sequence steps of the attack Scenario A

Step	Name	Description
1	Port Scan	Full, SYN,FIN, and UDP Scan
2	Web Crawling	bWAPP
3	Password Cracking	Brute force

The volume-based DDoS was performed based on three sequential steps as presented in Table 5. The first step of the volume-based DDoS attack is to execute the port scan attack (Full, SYN, FIN, and UDP Scan) simultaneously as shown in Table 4. Then, the next step is to launch the

APP-based DDoS attack using HTTP Slowloris DDoS attack as follows:

```
nmap --max-parallelism 750 -Pn
--script [SCRIPT_FILE]
--script-args http-slowloris.
runforever=true [IP-ADDRESS]
-p 80 -S [IP-ADDRESS]
```

Finally, executing the volume-based DDoS attack using the radware tool[2]. This scenario took an hour and three hosts (192.168.159.131, 192.168.159.14, and 192.168.159.16) were infected by the volume-based DDoS attack.

It should be mentioned here that the normal traffic was launched within the victim network during running both scenarios. Different protocols (HTTP, HTTPS, FTP, and ..etc) were used for the normal traffic to reflect the actual normal network behavior.

After implementing the multi-step attack scenarios A and B, six PCAP files were collected as follows:

- S0 presents (Normal traffic).
- S1 presents (Port Scan Traffic [Full, SYN, FIN, and UDP Scan]).
- S-A1 presents (App-based DDoS [HTTP Slowloris DDoS]).
- S-A2 presents (Volume-based DDoS [ICMP Flood]).
- S-B1 presents (Web Crawling).
- S-B2 presents (Password Cracking [Brute Force]).

The six PCAP files were processed using Wireshark[3]. Throughout the processing, we analyzed the timestamp of the network traffic (malicious and normal traffic) in order to label the network traffic. After processing these PCAP files, the generated dataset (MSCAD) contains 77 features (network parameters) with labels. These features were extracted using CIC-Flow-Meter as shown in Table 12 (see appendix). The PCAP files and the labeled-version are available on the MSCAD website[4].

The MSCAD contains 128,799 instances with 77 features (including the identifier features: Source Address, a Destination Address, Source Port, and Destination Port). Figure 3 shows the distribution of instances among the class labels. To evaluate the quality of the MSCAD, Firstly, two experiments were performed to build the detection models using the state-of-art machine learning algorithms. Secondly, the MSCAD was compared with other open-source and public datasets that were mentioned in Section 2 based on the latest keys criteria in [25, 30].

[2]https://www.radware.com/

[3]https://www.wireshark.org/

[4]The link will be available when the paper is published

4. Experiments and results

To evaluate the performance and robustness of the detection model for detecting different attack scenarios, we used the Geometric Mean (G-mean) and the Area Under Curve (AUC) measures. The reason for choosing these measures is that the MSCAD is a multi-class imbalanced data where the distribution of instances is skewed among the class labels as shown in Fig. 3. Thus, other measures such as recall, accuracy, or misclassification rate, do not accurately measure the detection model's performance. Besides, G-mean and AUC measures are highly sensitive to the rate of correctly classified instances of the majority and minority class labels [31, 32]. G-mean is one measure that used to evaluate the performance of the classifier (the detection model) when applied to an imbalanced dataset [31, 32]. Furthermore, the AUC presents the trade-off between sensitivity and specificity, which is considered as one of the important performance metrics when the dataset is imbalanced [1, 31, 32].

In this paper, two experiments were performed to evaluate the performance and robustness of the detection models as well as compare the performance of various detection models according to the G-mean and AUC performance metrics. In two experiments; two state-of-art machine learning algorithms were used; namely Decision Tree (DT) and Random-Forest (RF), to build a detection model to detect the attack for each instance in the testing dataset. In the first experiment, MSCAD (Multi-Class Unbalanced Data Set) was used to build and train the detection model without applying re-sampling techniques. In the second experiment, the MSCAD was modified using four re-sampling techniques; Synthetic Minority Oversampling Technique (SMOTE) [33], Borderline-SMOTE [34], Synthetic Minority Oversampling Technique based on Edited Nearest Neighbours (SMOTEENN) [35], and SMOTETomek [35], before training the detection model.

The re-sampling techniques are the data-level methods that are used to handle the skewed distribution of instances among class labels by changing the training dataset [36]. These techniques could be

Table 4
The port scan attack sequence steps

Running Scripts	Description
nmap -sT -mtu 32 -p 0-4000 {}P-ADDRESS{}	Fragmented TCP full scan for range of ports (0-4000)
nmap -sU -T4 -p 0-4000 {}P-ADDRESS{}	High-speed UDP scan for range of ports (0-4000)
nmap -sF -mtu 16 -p 0-4000 {}P-ADDRESS{}	Fragmented TCP FIN scan for range of ports (0-4000)
nmap -sS -Pn -A -S {}P-ADDRESS{} -p 0-1023 {}P-ADDRESS{} -e Eth4	Spoofed SYN scan with OS fingerprinting for range of ports (0-1023)

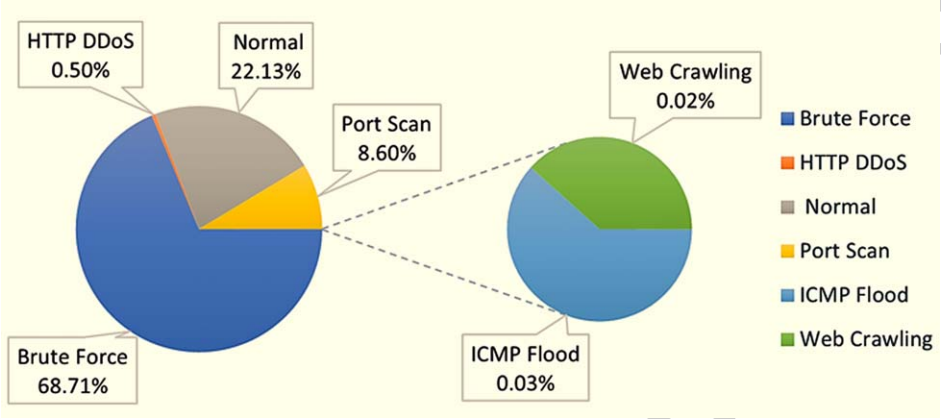


Fig. 3. The class labels' distribution.

Table 5
The sequence steps of the attack scenario B

Step	Name	Description
1	Port Scan	Full, SYN,FIN, and UDP Scan
2	APP-based DDoS	HTTP Slowloris DDoS attack
3	Volume-based DDoS	ICMP Flood

categorized either as oversampling, undersampling, or hybrid. In the oversampling technique, the synthetic instances are created for minority class labels based on the existing instances, such as SMOTE and Borderline-SMOTE. For the undersampling technique, the number of instances of majority class labels is reduced, such as ENN. The hybrid technique is the combination of oversampling and undersampling techniques such as SMOTEENN and SMOTETomek [37].

We used stratified K-Folds cross-validation with 5 folds to measure the detection model's performance. In the stratified K-Folds cross-validation technique, the dataset is split into k-folds where k-1 folds are used for training the detection model and 1-fold for testing the detection model. The aim behind using the stratified K-folds cross-validation technique is that the k-folds approximately have the same percentage of the instances for each class label.

As discussed at section 2.1, the MSCAD contains 128,799 instances with 77 features. In order to build a robust detection approach without any prior information about the source of the attack, the identifier features should be removed. After removing the identifier features, the MSCAD includes 66 features. This number of features is still considered as a large number of input parameters, as well not all of 66 features are relevant to detect the multi-step attack. Furthermore, the existence of irrelevant parameters could be affected the performance of the detection model [1, 11, 12]. Consequently, before training the detection model, three selection feature algorithms were applied to select the 10 best features that are more relevant to detect the multi-step attacks. These selection feature algorithms are the Correlation attribute evaluator (Correlation_Attr_Eval), InfoGain attributes evaluator (InfoGain_Attr_Eval), and GainRatio attribute evaluator (GainRatio_Attr_Eval). The aim behind using three different selection feature algorithms is to study and discover the impact of the selected shortlist features on the performance of the detection model. Tables 6, 7, and 8 show the best 10

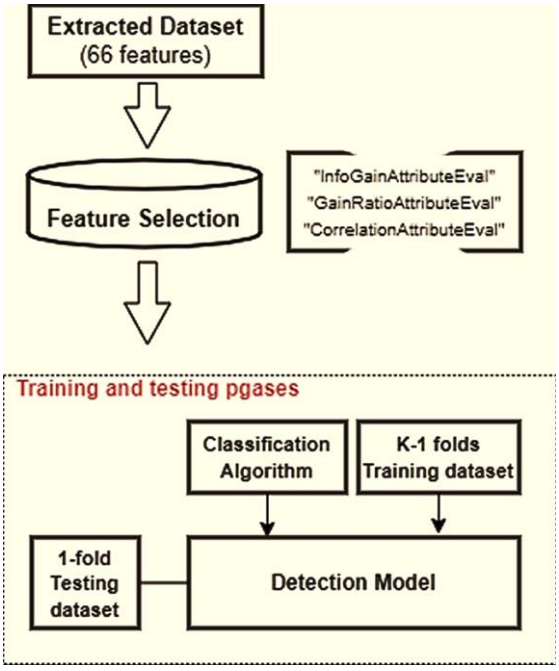


Fig. 4. Training and testing phases without resampling techniques.

Table 6
The Features selected by Correlation Attributes Evaluator

#	Feature	Rank
1	Flow_Duration	0.532
2	Fwd_IAT_Tot	0.491
3	Down_Up_Ratio	0.462
4	Flow_IAT_Max	0.412
5	Bwd_Pkt_Len_Std	0.381
6	Fwd_IAT_Max	0.38
7	Idle_Max	0.38
8	Idle_Mean	0.374
9	Pkt_Len_Std	0.366
10	Idle_Min	0.357

Table 7
The Features selected by Info Gain Attributes Evaluator

#	Feature	Rank
1	Init_Bwd_Win_Byts	1.14076
2	Bwd_Header_Len	1.02603
3	Fwd_Pkts/s	1.00705
4	Flow_IAT_Max	0.96144
5	Flow_Duration	0.95928
6	Bwd_Pkts/s	0.95286
7	Flow_Pkts/s	0.92127
8	Flow_IAT_Mean	0.85895
9	Fwd_Header_Len	0.84148
10	Flow_IAT_Min	0.82819

features together with their rank values that selected by Correlation_Attr_Eval, InfoGain_Attr_Eval, and GainRatio_Attr_Eval, respectively.

Table 8
The Features selected by Gain Ratio Attributes Evaluator

#	Feature	Rank
1	Init_Bwd_Win_Byts	0.48
2	Pkt_Len_Min	0.438
3	Active_Mean	0.42
4	Active_Max	0.42
5	Active_Min	0.415
6	Bwd_Pkt_Len_Min	0.391
7	Idle_Mean	0.388
8	Bwd_Header_Len	0.386
9	Fwd_Pkt_Len_Min	0.378
10	Idle_Max	0.378

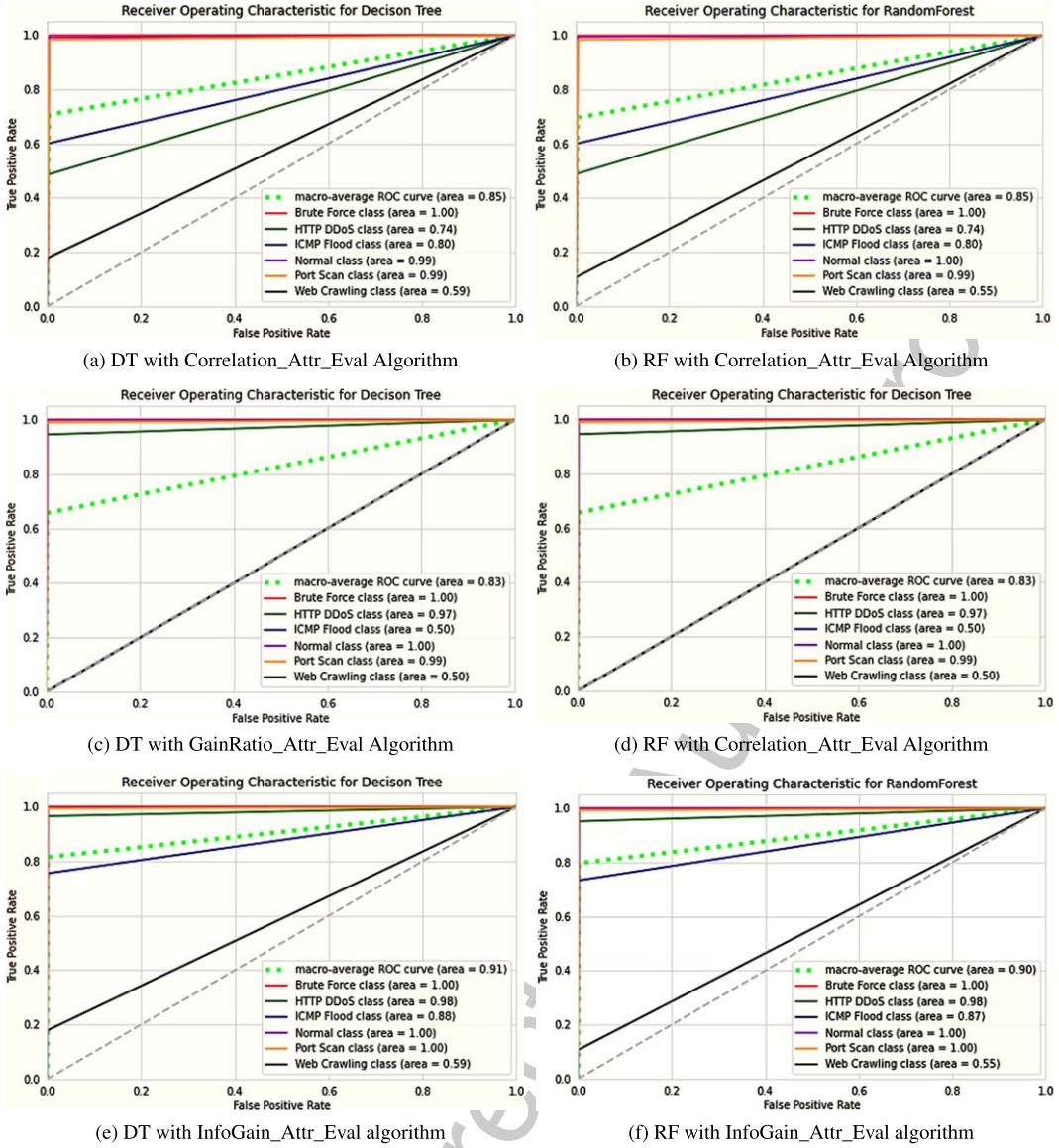
Table 9
The G-Mean Obtained by The DT and RF Detection Models Without Resampling

Features Selection	Decision Tree	Random Forest
InfoGain	0.71	0.65
Correlation	0.60	0.56
GainRatio	0.0	0.0

4.1. First experiment

In the first experiment, Decision Tree (DT) and RandomForest (RF) algorithms were applied to the MSCAD (without using the resampling techniques) for training and testing the detection models as shown in Fig. 4. Table 9 shows the overall performance of the two machine learning algorithms; DT and RF, with 5-folds according to the G-mean metric. From the results, the DT algorithm achieved the best G-mean with 0.71 when InfoGain_Attr_Eval was applied, compared to the performance of DT algorithm when GainRatio_Attr_Eval and Correlation_Attr_Eval algorithms were applied. Furthermore, the performance of the DT algorithm when InfoGain_Attr_Eval was applied, outperforms the performance of RF algorithms regardless of the feature selection algorithm that applied. Another conclusion from the results is that the DT and RF algorithms obtained the worst performance with 0.0 G-mean when GainRatio_Attr_Eval algorithm was applied.

Figures 5a to 5f show the ROC curve for each class label (attack and normal) achieved by DT and RF algorithms. As presented in Figs. 5a to 5f, the DT and RF algorithms obtained good macro-average AUC results in the case of applying three feature selection algorithms. However, DT and RF algorithms obtained a poor performance for detecting the web crawling attack, where DT algorithm with InfoGain_Attr_Eval obtained 0.59 as the best AUC for detecting the web crawling attack. Besides, the RF algorithm obtained 0.55 as the best AUC for detecting the web crawl-



(a) DT with Correlation_Attr_Eval Algorithm

(b) RF with Correlation_Attr_Eval Algorithm

(c) DT with GainRatio_Attr_Eval Algorithm

(d) RF with Correlation_Attr_Eval Algorithm

(e) DT with InfoGain_Attr_Eval algorithm

(f) RF with InfoGain_Attr_Eval algorithm

Fig. 5. ROC Curve Obtained by DT and RF Detection Models Without Resampling.

ing attack using the same feature selection algorithm (InfoGain_Attr_Eval). Another conclusion that can be drawn from these figures, the DT and RF algorithms somehow behave randomly for detecting the web crawling attack.

4.2. *Second experiment*

In order to obtain a better detection model's performance for detecting the minority class labels (minority attack), the skewed distribution of class labels in the MSCAD should be handled before training the detection models. In this experiment, various re-sampling techniques were applied to handle the imbalanced issue, which are SMOTE, Borderline-SMOTE, SMOTEENN, and SMOTETomek. Figure 6 illustrates the resampling technique that was applied to the MSCAD before starting training the detection model.

Table 10 shows the performance of the detection models according to the G-mean using both machine learning algorithms (DT and RF) with resampling techniques. As shown in the achieved results, the DT algorithm obtained the best G-mean with 0.83

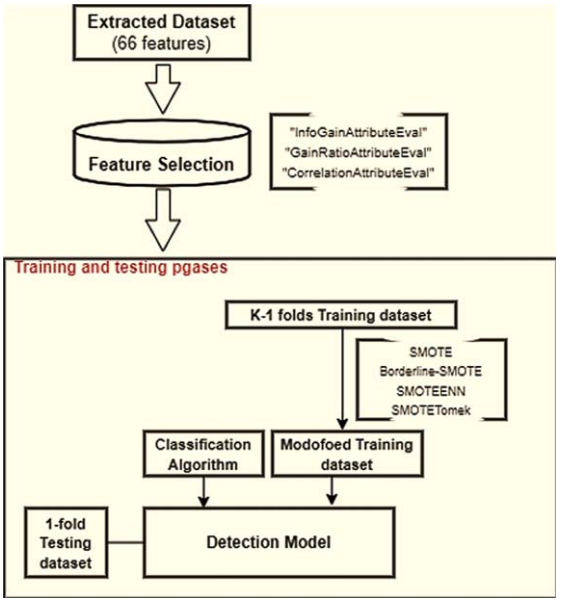


Fig. 6. Training and testing phases with resampling techniques.

Table 10
The G-Mean Obtained by The DT and RF Detection Models With Resampling

Methods	InfoGain	Correlation	GainRatio
DT with SMOTE	0.78	0.71	0.77
DT with Borderline-SMOTE	0.78	0.65	0.62
DT with SMOTEENN	0.79	0.76	0.56
DT with SMOTETomek	**0.83**	0.72	0.80
RF with SMOTE	0.79	0.71	0.77
RF with Borderline-SMOTE	0.78	0.69	0.00
RF with SMOTEENN	0.79	0.77	0.62
RF with SMOTETomek	0.82	0.74	0.78

when the SMOTETomek resampling technique and InfoGain_Attr_Eval algorithm were applied comparing with the other results in this table. To measure the detection model's performance for detecting the minority class labels (minority attacks) especially the web crawling attack label, the ROC curves obtained by the best detection model, DT algorithm with SMOTETomek and InfoGain_Attr_Eval, were drawn as shown in Fig. 7. From this Figure, we can see that the DT algorithm with SMOTETomek and InfoGain_Attr_Eval obtains the macro-average with 0.93 which is a little better than the macro-average AUC achieved by DT and RF algorithms without using the resampling techniques as shown in Figs. 5a to 5f. However, adapting the DT algorithm with SMOTETomek and InfoGain_Attr_Eval achieved a good AUC for detecting the web crawling attack which is 0.73.

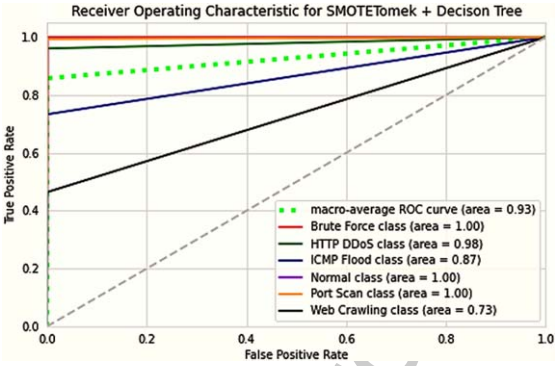


Fig. 7. The ROC Curves obtained by DT with SMOTETomek and InfoGain attribute evaluator.

5. Comparing with the public intrusion datasets

There are plausible contributions of generating intrusion datasets as discussed in Section 2. Nevertheless, some intrusion datasets have major shortcomings as mentioned in Section 2.1, such as not having the sequence steps of multi-step attack scenarios. Moreover, not including the recent cyber-attacks and excluding the recent network protocols.

Contrarily, the MSCAD handles the previous shortcomings and provides a suitable environment for training, testing, and evaluating different detection approaches of multi-step attacks because of having various attack scenarios. These scenarios were conducted based on sequential attacker steps. Moreover, The studied multi-step attack scenarios include the recent cyber-attacks and network protocols. To conduct a comprehensive comparison with other free open-source and public intrusion datasets discussed in Section 2, the latest keys criteria of the dataset evaluation framework [25, 30] were used as shown in Table 11 (see appendix).

The latest keys criteria summarized as follows:

- Labeled: this key indicates if the IDS dataset had a labeled version or not to distinguish the normal and malicious traffic.
- Format: this key indicates the type of dataset format (flow-based, packet-based, Bi-based or other formats).
- Year of Creation: this key presents the year of creation for the datasets.
- Meta-Data: this key is to investigate if the dataset is completely explained, and discussed. This include all information about the studied attacks and the test-bed environment.

- Attack Scenario: this key is to highlight if the dataset includes a multi-step attack scenario.
- Kind of Traffic: this key to identify the type of dataset traffic (real, emulated, synthetic).
- Complete Network Configuration: this key to present if the dataset traffic generated from complete network topology (switches, routers, firewall).
- Complete Interaction: this key to indicate if the dataset traffic generated from two or more networks beside internet service.
- Protocols: this key to highlight the types of protocols that are used in the generated dataset.
- Attack Diversity: this key presents if the dataset includes the recent cyber-attacks such as DoS, DDoS, Brute force, Web-based, Scan, Bot, and infiltration. The attack diversity is presented based on McAfee security threat report (Q1), 2019 [38].
- Heterogeneity: this key to indicate if the traffic of the dataset collected from all infected victims and switches.

6. Conclusion

This paper has introduced a new benchmark dataset for intrusion detection systems (MSCAD). The MSCAD includes two attack scenarios; the first scenario is a password cracking attack where the attacker aimed to execute a password cracking attack (Brute force) on any client within the victim network. The second attack scenario is a volume-based DDoS attack where the attacker aimed to execute volume-based DDoS on any client within the victim network. The MSCAD is available in two formats (PCAP's and CSV's). The PCAP files are available to be processed for offline-mode (replay mode) along with an open-source intrusion detection system to analyze the alert correlations during the malicious traffic. On the other hand, the generated multi-step attack dataset was completely labeled with 77 features (network parameters) that were extracted using CIC-Flow-Meter.

The MSCAD was evaluated in different manners. Firstly, the flow-based features were examined and investigated to select the best shortlist features using three feature selection algorithms (Correlation Attributes Evaluator, Info Gain Attributes Evaluator, Gain Ratio Attribute Evaluator). Then, the skewed distribution of class labels (attacks and normal behavior) was handled using resampling techniques to improve the detection model's performance. Afterwards, two machine learning algorithms (Random Forest, and Decision Tree) were applied to the proposed dataset to evaluate the performance and accuracy of the selected features. The experimental results demonstrate that the decision tree algorithm with SMOTETomek and InfoGain attributes evaluator achieved the best performance with G-mean 0.83.

Finally, the MSCAD was compared with other public and open-source intrusion datasets for free based on the latest key criteria of the dataset evaluation framework. The results showed that the MSCAD successfully passed twelve major criteria. Moreover, the presented multi-step attack dataset can be a promising intrusion dataset for training, testing, and evaluation of detection methods for multi-step attack detection.

References

[1] Mohammad Almseidin, Imre Piller, Mouhammd Al-Kasassbeh and Szilveszter Kovacs, Fuzzy automaton as a detection mechanism for the multi-step attack, *International Journal on Advanced Science, Engineering and Information Technology* **9**(2), 2019.

[2] Mohammad Almseidin, Mouhammd Al-Kasassbeh and Szilveszter Kovacs, Detecting slow port scan using fuzzy rule interpolation, In *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pages 1–6. IEEE, 2019.

[3] Da-peng MAN, Xue-zhen LI, Wu YANG, Wei WANG and Shi-chang XUAN. A multi-step attack recognition and prediction method via mining attacks conversion frequencies, *Int J Wirel Microw Technol (IJWMT)* **2**(2) (2012), 20–25.

[4] Shigen Shen, Keli Hu, Longjun Huang, Hongjie Li, Risheng Han and Qiying Cao, Quantal response equilibrium-based strategies for intrusion detection in wsns, *Mobile Information Systems* **2015** (2015).

[5] Yanxue Zhang, Dongmei Zhao and Jinxing Liu, The application of baum-welch algorithm in multistep attack, *The Scientific World Journal* **2014** (2014).

[6] Mohammad Almseidin, Jamil Al-Sawwa and Mouhammd Alkasassbeh, Anomaly-based intrusion detection system using fuzzy logic, In *2021 International Conference on Information Technology (ICIT)*, pages 290–295, IEEE, 2021.

[7] Mouhammd Al-Kasassbeh, Mohammad Almseidin, Khaled Alrfou and Szilveszter Kovacs, Detection of iot-botnet attacks using fuzzy rule interpolation, *Journal of Intelligent & Fuzzy Systems* **39** (2020), 421–431.

[8] Shigen Shen, Haiping Zhou, Sheng Feng, Longjun Huang, Jianhua Liu, Shui Yu and Qiying Cao, Hsird: A model for characterizing dynamics of malware diffusion in heterogeneous wsns, *Journal of Network and Computer Applications* **146** (2019), 102420.

[9] Shigen Shen, Longjun Huang, Haiping Zhou, Shui Yu, En Fan and Qiying Cao, Multistage signaling game-based optimal detection strategies for suppressing malware diffusion

in fogcloud-based iot networks, *IEEE Internet of Things Journal* **5**(2) (2018), 1043–1054.

[10] Haiping Zhou, Shigen Shen and Jianhua Liu, Malware propagation model in wireless sensor networks under attack-defense confrontation, *Computer Communications* **162** (2020), 51–58.

[11] Mohammad Almseidin, Mouhammd Al-kasassbeh and Szilveszter Kovacs, Fuzzy rule interpolation and snmp-mib for emerging network abnormality, *International Journal on Advanced Science, Engineering and Information Technology* **9**(3) (2019), 735–744.

[12] Mohammad Almseidin and Szilveszter Kovacs, Intrusion detection mechanism using fuzzy rule interpolation, *Journal of Theoretical and Applied Information Technology* **96**(16) (2018), 5473–5488.

[13] Jianhua Liu, XinWang, Shigen Shen, Guangxue Yue, Shui Yu and Minglu Li, A bayesian q-learning game for dependable task offloading against ddos attacks in sensor edge cloud, *IEEE Internet of Things Journal* **8**(9) (2020), 7546–7561.

[14] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes and Andreas Hotho, A survey of network-based intrusion detection data sets, *Computers and Security* **86** (2019), 147–167.

[15] Ansam Khraisat, Iqbal Gondal, Peter Vamplew and Joarder Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, *Cybersecurity* **2**(1) (2019), 20.

[16] A taxonomy of network threats and the effect of current datasets on intrusion detection systems, *IEEE Access* **8** (2020), 104650–104675.

[17] S. Hettich, The uci kdd archive, 1999. http://kdd.ics.uci.edu

[18] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, A detailed analysis of the kdd cup 99 data set, In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. IEEE, 2009.

[19] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee and Ali A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, *Computers & Security* **31**(3) (2012), 357–374.

[20] Mohammad Almseidin, Maen Alzubi, Mouhammd Alkasassbeh and Szilveszter Kovacs, Applying intrusion detection algorithms on the kdd-99 dataset, *Production Systems and Information Engineering* **8** (2019), 51–67.

[21] Paul Hick, Emile Aben, Kc Claffy and Josh Polterock, the caida ddos attack 2007 dataset, 2007.

[22] Francesco Gringoli, Luca Salgarelli, Maurizio Dusi, Niccolo Cascarano, Fulvio Risso and K.C. Claffy, Gt: picking up the truth from the ground for internet traffic, *ACM SIGCOMM Computer Communication Review* **39**(5) (2009), 12–18.

[23] Mouhammd Alkasassbeh, Ghazi Al-Naymat, Ahmad Hassanat and Mohammad Almseidin, Detecting distributed denial of service attacks using data mining techniques, *International Journal of Advanced Computer Science and Applications* **7**(1) (2016), 436–445.

[24] Gideon Creech and Jiankun Hu, A semantic approach to hostbased intrusion detection systems using contiguousand discontiguous system call patterns, *IEEE Transactions on Computers* **63**(4) (2013), 807–819.

[25] Iman Sharafaldin, Arash Habibi Lashkari and Ali A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, In *ICISSP*, pages 108–116, 2018.

[26] Ranjit Panigrahi and Samarjeet Borah, A detailed analysis of cicids2017 dataset for designing intrusion detection systems, *International Journal of Engineering & Technology* **7**(3.24) (2018), 479–482.

[27] Monowar H. Bhuyan, Dhruba K. Bhattacharyya and Jugal K. Kalita, Towards generating real-life datasets for network intrusion detection, *IJ Network Security* **17**(6) (2015), 683–701.

[28] Javed Asharf, Nour Moustafa, Hasnat Khurshid, Essam Debie, Waqas Haider and Abdul Wahab, A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions, *Electronics* **9**(7) (2020), 1177.

[29] Shigen Shen, Haiping Zhou, Sheng Feng, Longjun Huang, Jianhua Liu, Shui Yu and Qiying Cao, Hsird: A model for characterizing dynamics of malware diffusion in heterogeneous wsns, *Journal of Network and Computer Applications* **146** (2019), 102420.

[30] Amirhossein Gharib, Iman Sharafaldin, Arash Habibi Lashkari and Ali A. Ghorbani, An evaluation framework for intrusion detection dataset, In *2016 International Conference on Information Science and Security (ICISS)*, pages 1–6. IEEE, 2016.

[31] J. Al-Sawwa and S.A. Ludwig, A cost-sensitive centroid-based differential evolution classification algorithm applied to cancer data sets, In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2514–2521, 2019.

[32] A. Puri and M.K. Gupta, Comparative analysis of resampling techniques under noisy imbalanced datasets, In *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, volume **1**, pages 1–5, 2019.

[33] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer, Smote: synthetic minority oversampling technique, *Journal of Artificial Intelligence Research* **16** (2002), 321–357.

[34] Hui Han, Wen-Yuan Wang and Bing-Huan Mao, Borderlinesmote: A new over-sampling method in imbalanced data sets learning, In *Advances in Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[35] Gustavo E.A.P.A. Batista, Ronaldo C. Prati and Maria Carolina Monard, A study of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor Newsl* **6**(1) (2004), 20–29.

[36] Ajinkya More, Survey of resampling techniques for improving classification performance in unbalanced datasets, 2016.

[37] Jun Ye, Multiple attribute decision-making methods based on the expected value and the similarity measure of hesitant neutrosophic linguistic numbers, *Cognitive Computation* **10**(3) (2018), 454–463.

[38] R. Samani and G. Davis, Mcafee threat report q1, 2019.

Appendix

Table 11
The Extracted Features

No	Feature Name	Description
1	Flow duration	Duration of the flow in Microsecond
2	Total Fwd Packet	Total packets in the forward direction
3	Total Bwd Packets	Total packets in the backward direction
4	Total Length of Fwd Packet	Total size of packet in forward direction
5	Total Length of Bwd Packet	Total size of packet in backward direction
6	Fwd Packet Length Min	Minimum size of packet in forward direction
7	Fwd Packet Length Max	Maximum size of packet in forward direction
8	Fwd Packet Length Mean	Mean size of packet in forward direction
9	Fwd Packet Length Std	Standard deviation size of packet in forward direction
10	Bwd Packet Length Min	Minimum size of packet in backward direction
11	Bwd Packet Length Max	Maximum size of packet in backward direction
12	Bwd Packet Length Mean	Mean size of packet in backward direction
13	Bwd Packet Length Std	Standard deviation size of packet in backward direction
14	Flow Bytes/s	Number of flow bytes per second
15	Flow Packets/s	Number of flow packets per second
16	Flow IAT Mean	Mean time between two packets sent in the flow
17	Flow IAT Std	Standard deviation time between two packets sent in the flow
18	Flow IAT Max	Maximum time between two packets sent in the flow
19	Flow IAT Min	Minimum time between two packets sent in the flow
20	Fwd IAT Min	Minimum time between two packets sent in the forward direction
21	Fwd IAT Max	Maximum time between two packets sent in the forward direction
22	Fwd IAT Mean	Mean time between two packets sent in the forward direction
23	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
24	Fwd IAT Total	Total time between two packets sent in the forward direction
25	Bwd IAT Min	Minimum time between two packets sent in the backward direction
26	Bwd IAT Max	Maximum time between two packets sent in the backward direction
27	Bwd IAT Mean	Mean time between two packets sent in the backward direction
28	Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
29	Bwd IAT Total	Total time between two packets sent in the backward direction
30	Fwd PSH flags	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
31	Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
32	Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
33	Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
34	Fwd Header Length	Total bytes used for headers in the forward direction
35	Bwd Header Length	Total bytes used for headers in the backward direction
36	FWD Packets/s	Number of forward packets per second
37	Bwd Packets/s	Number of backward packets per second
38	Packet Length Min	Minimum length of a packet
39	Packet Length Max	Maximum length of a packet
40	Packet Length Mean	Mean length of a packet
41	Packet Length Std	Standard deviation length of a packet
42	Packet Length Variance	Variance length of a packet
43	FIN Flag Count	Number of packets with FIN
44	SYN Flag Count	Number of packets with SYN
45	RST Flag Count	Number of packets with RST
46	PSH Flag Count	Number of packets with PUSH
47	ACK Flag Count	Number of packets with ACK
48	URG Flag Count	Number of packets with URG
49	CWR Flag Count	Number of packets with CWR
50	ECE Flag Count	Number of packets with ECE
51	down/Up Ratio	Download and upload ratio
52	Average Packet Size	Average size of packet
53	Fwd Segment Size Avg	Average size observed in the forward direction
54	Bwd Segment Size Avg	Average number of bytes bulk rate in the backward direction
55	Fwd Bytes/Bulk Avg	Average number of bytes bulk rate in the forward direction
56	Fwd Packet/Bulk Avg	Average number of packets bulk rate in the forward direction

(*Continued*)

Table 11
(*Continued*)

No	Feature Name	Description
57	Fwd Bulk Rate Avg	Average number of bulk rate in the forward direction
58	Bwd Bytes/Bulk Avg	Average number of bytes bulk rate in the backward direction
59	Bwd Packet/Bulk Avg	Average number of packets bulk rate in the backward direction
60	Bwd Bulk Rate Avg	Average number of packets bulk rate in the backward direction
61	Bwd Bulk Rate Avg	Average number of bulk rate in the backward direction
62	Subflow Fwd Packets	The average number of packets in a sub flow in the forward direction
63	Subflow Fwd Bytes	The average number of bytes in a sub flow in the forward direction
64	Subflow Bwd Packets	The average number of packets in a sub flow in the backward direction
65	Subflow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
66	Fwd Init Win bytes	The total number of bytes sent in initial window in the forward direction
67	Bwd Init Win bytes	The total number of bytes sent in initial window in the backward direction
68	Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
69	Fwd Seg Size Min	Minimum segment size observed in the forward direction
70	Active Min	Minimum time a flow was active before becoming idle
71	Active Mean	Mean time a flow was active before becoming idle
72	Active Max	Maximum time a flow was active before becoming idle
73	Active Std	Standard deviation time a flow was active before becoming idle
74	Idle Min	Minimum time a flow was idle before becoming active
75	Idle Mean	Mean time a flow was idle before becoming active
76	Idle Max	Maximum time a flow was idle before becoming active
77	Idle Std	Standard deviation time a flow was idle before becoming active

Table 12
MSCAD vs. Other Intrusion Datasets

Dataset	Labeled	Format	Year of Creation	Meta Data	Attack Scenario	Kind of Traffic	Network Configuration	Complete Interaction	HTTP	HTTPS	SSH	FTP	Attack Diversity	Hetero-geneity
KDDcup98	Yes	other	1998	No	No	emulated	No	Yes	Yes	No	Yes	Yes	No	No
NSL-KDD	Yes	other	1998	No	No	emulated	No	Yes	Yes	No	Yes	Yes	No	No
DARPA	No	packet, Logs	1998-1999	Yes	Yes	emulated	No	Yes	Yes	No	Yes	Yes	No	No
CAIDA	No	Flow	2007	No	No	real	Yes	No	Yes	Yes	Yes	No	No	No
UNIBS	No	Flow	2009	No	No	real	Yes	Yes	Yes	No	Yes	Yes	No	No
ISCX 2012	Yes	packet, bi, flow	2012	Yes	Yes	emulated	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
DDoS 2016	Yes	packet,	2016	No	No	synthetic	No	No	Yes	No	Yes	No	No	No
ADFA-LD	Yes	other	2015	Yes	Yes	emulated	No	Yes	Yes	No	Yes	Yes	No	No
ADFA-WD	Yes	other	2015	Yes	Yes	emulated	No	Yes	Yes	No	Yes	Yes	No	No
CIC-IDS 2017	Yes	packet, bi, flow	2017	Yes	No	emulated	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TUIDS	Yes	packet, bi, flow	2011	No	No	emulated	No	No	Yes	Yes	Yes	Yes	No	No
MSCAD	Yes	flow	2021	Yes	Yes	real	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes