An Assessment of Intrusion Detection using Machine Learning on Traffic Statistical Data

Qianru Zhou [1], Rongzhen Li [2], Lei Xu [2], Hongyi Zhu [2], and Wanli Liu [2]

[1]Nanjing University of Science and Technology
[2]Affiliation not available

October 30, 2023

Abstract

Detecting Zero-Day intrusions has been the goal of Cybersecurity, especially intrusion detection for a long time. Machine learning is believed to be the promising methodology to solve that problem, numerous models have been proposed but a practical solution is still yet to come, mainly due to the limitation caused by the out-of-date open datasets available. In this paper, we propose an approach for Zero-Day intrusion detection based on machine learning, using flow-based statistical data generated by CICFlowMeter as training dataset. The machine learning classification model used is selected from eight most popular classification models, based on their cross validation results, in terms of precision, recall, F1 value, area under curve (AUC) and time overhead. Finally, the proposed system is tested on the testing dataset. To evaluate the feasibility and efficiency of tested models, the testing datasets are designed to contain novel types of intrusions (intrusions have not been trained during the training process). The normal data in the datasets are generated from real life traffic flows generated from daily use. Promising results have been received with the accuracy as high as almost 100%, false positive rate as low as nearly 0%, and with a reasonable time overhead. We argue that with the proper selected flow based statistical data, certain machine learning models such as MLP classifier, Quadratic discriminant analysis, K-Neighbor classifier have satisfying performance in detecting Zero-Day attacks.

An Assessment of Intrusion Detection using Machine Learning on Traffic Statistical Data

Qianru Zhou*, *Member, IEEE,* Rongzhen Li*,Xu Lei†, *Member, IEEE,* Hongyi Zhu,Wanli Liu†

***Abstract*—Detecting Zero-Day intrusions has been the goal of Cybersecurity, especially intrusion detection for a long time. Machine learning is believed to be the promising methodology to solve that problem, numerous models have been proposed but a practical solution is still yet to come, mainly due to the limitation caused by the out-of-date open datasets available. In this paper, we propose an approach for Zero-Day intrusion detection based on machine learning, using flow-based statistical data generated by CICFlowMeter as training dataset. The machine learning classification model used is selected from eight most popular classification models, based on their cross validation results, in terms of precision, recall, F1 value, area under curve (AUC) and time overhead. Finally, the proposed system is tested on the testing dataset. To evaluate the feasibility and efficiency of tested models, the testing datasets are designed to contain novel types of intrusions (intrusions have not been trained during the training process). The normal data in the datasets are generated from real life traffic flows generated from daily use. Promising results have been received with the accuracy as high as almost 100%, false positive rate as low as nearly 0%, and with a reasonable time overhead. We argue that with the proper selected flow based statistical data, certain machine learning models such as *MLP classifier, Quadratic discriminant analysis, K-Neighbor classifier* have satisfying performance in detecting Zero-Day attacks.**

***Index Terms*—Intrusion detection, Zero-Day attacks, cybersecurity, machine learning, CICFlowMeter**

I. Introduction

With the novel cyber attacks keep emerging, and the rapid extension of new communication protocols, which not only encrypts the user payload data but also scrambles the basic packet header information such as IP address and Port number [1], traditional intrusion detection methodologies which relays on these packet header information in finding and matching the patterns of intrusions are gradually losing their effectiveness [2–5]. Thus adopting machine learning technologies to detect intrusions are more and more believed to be the future solution for intrusion detection [3–12]. As a technology of Artificial Intelligence, machine learning is well known by its capability to grasp hidden patterns from massive datasets and provide accurate prediction.

The performance of a machine learning algorithm largely depends on the dataset it is trained on [6]. The majority of current machine learning based intrusion detection approaches are trained with DARPA 98/99[1], KDD CUP 99[2] datasets. However, the use of these datasets has become a serious issue and an increasing number of researchers recommend against their use [6, 12–16]. The CSE-CIC-IDS 2018 dataset[3] collected on one of Amazonś AWS LAN network (thus also known as the CIC-AWS-2018 Dataset) by the Canadian Institute of Cybersecurity (CIC) has gaining attention [17]. Besides the straightforward TCP/IP level traffic information such as IP address and port number, CIC-AWS datasets provides statistical traffic information based on flow, calculated by the network flow generator and analyser developed by CIC – CICFlowMeter.

There are six different intrusion scenarios in the dataset, Brute-force, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside, with a total of 14 types of intrusions, namely, *Botnet attack, FTP-BruteForce, SSH-BruteForce, BruteForce-Web, BruteForce-XSS, SQL Injection, DDoS-HOIC attack, DDoS-LOIC-UDP attack, DDoS-LOIC-HTTP attacks, Infiltration, DoS-Hulk attack, DoS-SlowHTTPTest attack, DoS-GoldenEye attack, and DoS-Slowloris attack*. All the data are fully labeled, describing the statistical features of the traffic, e.g., flow duration, number of packets, number of times a certain flag was set in packets, total bytes used for the header in an upward flow, etc. The raw record of network traffic and event logs are also provided in CIC-AWS-2018 Dataset , although they are not discussed in this paper.

Although the intrusion detection studies using CIC-AWS-2018 Dataset has not yet been much reported, there are plenty research work has been done on an earlier version of CIC-AWS-2018 Dataset , CICIDS2017 [17–19]. Radford et.al [18] has applied unsupervised learning on CICIDS2017. The creator of CICIDS2017 dataset, Sharafaldin et.al from the CIC institute use RandomForestRegression to choose the top four features that can best describe each attacks, and apply machine learning based intrusion detection for them [17].

The contribution of this paper is threefold. We present the design of the proposed intrusion detection system. After data laundry and feature selection, we evaluated the performance of eight most common machine learning classifiers on the benchmark of CIC-AWS-2018 Dataset with the metrics of precision, recall, f1-score, and the time overhead.

The paper is organized as follows. Section II-A provides detailed introduction of the datasets used in this paper, including

School of Compute Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. email: (xulei_marus@126.com; 18951768998@163.com)

*authors contribute equally to this paper.

† corresponding author.

Manuscript received April 19, 2005; revised August 26, 2015.

[1]https://www.ll.mit.edu/r-d/datasets

[2]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3]https://registry.opendata.aws/cse-cic-ids2018/

the CIC-AWS-2018 Dataset which is the training dataset, and the Zero-Day intrusions collected online and benign data collected in real research production environment. The procedure of data laundry adopted in this paper is also presented. Section II-C describes the machine learning methodology adopted in this paper. The evaluation experiments and the results are analyzed in detail in Section III, including the cross validation process and results, and the testing results. Finally, Section IV summaries the paper and provides our vision for future work.

II. Proposed Approach

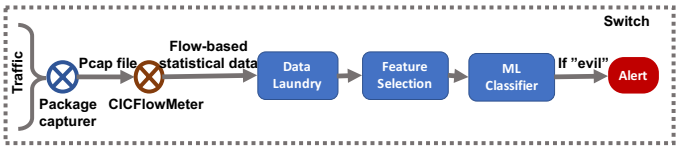


Fig. 1. The work flow of the proposed intrusion detection system.

A. Dataset

The datasets used in this paper are collected from three main origins, the CIC-AWS-2018 Dataset , benign datasets collected from the authors' daily production network, novel intrusion datasets collected from of network security experts' blogs or websites.

1) CIC-AWS-2018 Dataset : The performance of a machine learning algorithm is largely depends on the dataset it is trained on [6]. The majority of current machine learning based intrusion detection approaches are trained with DARPA 98/99[4], KDD CUP 99[5] datasets. However, these datasets are seriously out of date and an increasing number of researchers recommend against their use [6, 12–16]. The CSE-CIC-IDS 2018 dataset[6] collected on one of Amazonś AWS LAN network (thus also known as the CIC-AWS-2018 Dataset) by the Canadian Institute of Cybersecurity (CIC) has gaining attention [17]. Besides the straightforward TCP/IP level traffic information such as IP address and port number, CIC-AWS datasets provides 84 features that are statistical traffic information based on flow, calculated by the network flow generator and analyzer CICFlowMeter.

There are six different intrusion scenarios in the dataset, Brute-force, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside, with a total of 14 types of intrusions, namely, *Botnet attack, FTP-BruteForce, SSH-BruteForce, BruteForce-Web, BruteForce-XSS, SQL Injection, DDoS-HOIC attack, DDoS-LOIC-UDP attack, DDoS-LOIC-HTTP attacks, Infiltration, DoS-Hulk attack, DoS-SlowHTTPTest attack, DoS-GoldenEye attack, and DoS-Slowloris attack*. All the data are fully labeled, describing the statistical features of the traffic, e.g., flow duration, number of packets, number of times a certain flag was set in packets, total bytes used for the header in an upward flow, etc. The raw record of network traffic and event logs are also provided in CIC-AWS-2018 Dataset , although they are not discussed in this paper.

The features in CIC-AWS dataset are described in Table I. There are 80 features in the dataset, providing statistical information of the flows from both uplink and downlink. Comparing to the straightforward TCP/IP traffic header informations provided by the previous datasets, like DARPA and KDD, it is widely believed that the statistics based on flow could provide more useful information for intrusion detection [20].

2) Collected Dataset: We use real life traffic data as the source of test dataset. The test datasets are collected from mainly two different sources, the benign data and intrusion data.

Benign data are collected from our real-life online surfing traffic collected in a typical research production environment. It is generated during the following daily online activities: emailing, searching (mainly on Google), reading news, watching video (through Netflex and Youtube), downloading papers from Google Scholar.

The data are collected for a week on our office desktops in a research daily routine environment, and then converted into flow-based statistical dataset consisting of 12,681 MB.

3) Intrusion data from Web:: To evaluate the ability of the machine learning models in detecting an attack that it has not seen before (or in other word, Zero-Day attacks), we collect novel real-life attacks traffic data containing eight new attack types with no repetitive with the training CIC-AWS-2018 Dataset . This dataset is collected from most recent real life attacks or abnormal traffic that humans failed to detect and prevent, most of them are still active till nowadays, such as `ransom malware`, `DDoS Bot'a Darkness`, `Google doc macadocs`, and `Bitcoin Miner`(this is more like abnormal traffic rather than intrusions to many people).

B. Data Laundry

The following steps are adopted to laundry the CIC-AWS-2018 Dataset .

- Delete noisey features;
- Format data into standard datatype;
- To reduce the size of the datasets, reduce the unnecessary accuracy of the float numbers by dropping digits after the decimal point;
- Replace noisy, machine unprocessable chars by underline _ ;
- Replace *"Infinity"* and *"NaN"* value with suitable numbers.

After the laundry, the total size of training dataset has dropped from 6,886 MB to 4 MB, without losing valuable information.

Feature selection and ranking are crucial for machine learning. The removal of useless features enhances the accuracy and decreases the computation time, and therefore achieve higher performance [21].

[4]https://www.ll.mit.edu/r-d/datasets
[5]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[6]https://registry.opendata.aws/cse-cic-ids2018/

TABLE I
FEATURES USED IN CIC-AWS DATASET

Feature	Description	Type
Label	indicate whether the traffic is malicious or not, e.g., benign, SQL-Injection, etc.	string
Dst Port	Destination port number	integer
Protocol	Protocol	integer
TimeStamp	Time Stamp of the flow	string
Flow Duration	Flow duration	integer
Tot Fwd/Bwd Pkts	Total packets in forward/backward directions	integer
TotLen Fwd/Bwd Pkts	Total size of packets in forward/backward directions	integer
Fwd/Bwd Pkt Len Max/Min/Mean/Std	Maxi/Mini/Average/Std. Dev. size of package in forward/backward directions	integer
Flow Byts/s & Flow Pkts/s	Flow byte rate, i.e., number of packets per seconds	float64
Flow IAT Mean/Std/ Max/Min	Average/Std. Deviation/Maxi/Mini time between two flows	float64
Fwd/Bwd IAT Tot/Mean/ Std/Max/Min	Total/Average/Std. Deviation/Maxi/Mini time between two packets in forward/backward directions	float64
Fwd/Bwd PSH/URG Flags	Number of times the PSH/URG flag was set in packets in forward/backward direction	integer
Fwd/Bwd Header Len	Total bytes used for header in forward/backward direction	integer
Fwd/Bwd Pkts/s	Number of forward/backward packets per second	float64
Pkt Len Min/Max/Mean/Std	Maxi/Mini/Average/Std. Dev. length of a flow	integer
Pkt Len Var	Mini inter-arrival time of packet	float64
FIN/SYN/RST/PUSH/ACK/ URG/CWE/ECE Flag Cnt	Number of packets with FIN/SYN/RST/PUSH/ACK/URG/CWE/ECE	integer
Down/Up Ratio	Download/upload ratio	integer
Pkt Size Avg	Average size of packets in forward/backward direction	float64
Fwd/Bwd Seg Size/Byts/b/Blk Rate Avg	Average number of bulk rate/bytes bulk rate/packets bulk rate in forward/backward directions	float64
Subflow Fwd/Bwd Pkts/Byts	The average number of bytes/packets in a sub flow in forward/backward direction	integer
Init Fwd/Bwd Win Byts	Number of bytes sent in initial window in forward/backward directions	integer
Fwd Act Data Pkts	Number of packets with at least 1 byte of TCP data payload in forward	integer
Fwd Seg Size Min	Minimum segment size observed in forward	integer
Active Mean/Std/Max/Min	Maxi/Mini/Average/Std. Dev. a flow was active before becoming idle	float64
Idle Mean/Std/Max/Min	Maxi/Mini/Average/Std. Dev. a flow was idle before becoming active	float64

TABLE II
ATTACK TYPES INCLUDED IN TEST DATASET

Attack Type	Description
`Bitcoin Miner`	Traffic generated during Bitcoin mining, maybe not a typical attack, but is treated as traffic blocker in production network.
`Drowor worm`	A virus in Windows operation system that infects portable executable (PE) files, such as those with EXE, DLL, and SYS files. It stops security processes from running, and overwrites some of their code, which means that you may have to reinstall affected security programs[7].
`Nuclear ransomware`	A new version of file-encrypting virus that actively spreads in Hungary, Italy, and Iran. Crypto-malware uses a combination of RSA and AES encryption and appends .[black.world@tuta.io].nuclear extension. Criminals provide a ransom note in HELP.hta file and ask to contact black.world@tuta.io for more information.
`False content injection`	Some network operators inject false content into users' network traffic, the injected packets have identical IP address, port number, and TCP sequence numbers, but different payload.
`Ponmocup trojan`	A trojan in Windows operation system, which tries to download other malware from the Internet.
`DDoS Bot'a Darkness`	Containing four types of DDoS attacks, namely HTTP flood, ICMP flood, ping, SYN flood and UDP flood. Still under active development by Russian malware coders.
`Google doc macadocs`	A new variant of the Macadocs malware to be using Google docs as a proxy server and not connecting to a command and control (C&C) server directly.
`ZeroAccess`	A Trojan horse computer malware that affects Microsoft Windows operating systems. It is used to download other malware on an infected machine from a botnet while remaining hidden using rootkit techniques.

C. Machine Learning Classifiers

In our problem model, the task is that given a set of statistic information of a flow, identify whether this flow is benign traffic or intrusion, by learning a set of already labelled data containing both benign and intrusion traffic, that makes our problem a *supervised classification* problem. To evaluate the performance of current popular classification models, we choose the eight most commonly used machine learning classification models, and train them with the training dataset, evaluate their performance with the criterias of *precision*, *recall*, *F1 score*, *AUC* and *time expense*.

The supervised machine learning classification models we evaluated in this paper are listed below.

- Random forest classifier
- Gaussian naive bayes classifier
- Decision tree classifier
- Multi-layer Perceptron (MLP) classifier
- K-nearest neighbors classifier
- Quadratic discriminant analysis classifier
- Support vector classifier

- Gradient boosting classifier

The performance of each classification model will be analyzed in detail during the cross validation section below in Section III.

III. Evaluation

This section presents the evaluation experiments setup and the evaluation results of intrusion detections. Two experiments are carried out to answer the following questions respectively:

Q1: Can all these types of intrusions be detected with these common machine learning models using the statistical features calculated by CICFlowMeter? How accurate is the detection, in terms of false positive rate (FPR), true negative rate (TNR), and area under the curve (AUC)? What is the time expense for these detections?

Q2: Which one of these machine learning models is able to detect intrusions that have never met before (have not been trained)? How accurate is the detection in terms of FPR, TNR, and AUC? What is the time expense?

To answer these questions, two experiments are designed in the following way respectively:

Experiment 1: *Datasets of each type of intrusions are divided into training set and testing set randomly (80% for training set, and 20% for testing set), and apply detection with each one of these machine learning models in turn. Evaluate the results in terms of accuracy and time expense.*

Experiment 2: *Apply detection tests for intrusions never appeared in training datasets, on each one of the machine learning models. Evaluate results from experiments run multiple times while gradually increase the diversity of intrusion types in the training dataset, in terms of accuracy and time expense.*

A. Environment

The software tool used in the evaluation experiments is *python3.8*, *PyTorch1.7*[8], *sklearn7*[9], *numpy8*[10], and *pandas9*[11]. All the evaluation experiments are carried on a HP laptop operating on Win10 64bit system, with 2.30 GHz on i5-6200U CPU, 8 GB memory and 1 TB hardware disk.

B. Experiment 1: Cross Validation for Each Intrusion

To evaluate the fitness of each machine learning classification models for each types of intrusion respectively, we run a 5-fold cross validation with the 8 most popular machine learning models, on each of the eleven types of attacks in the CIC-AWS-2018 Dataset . The results are shown in detail in Table III, IV, V, VI, VII and VIII, where the results of `SSH-BruteForce Attack`, `DDoS-HOIC Attack`, `DDOS-LOIC-UDP Attack`, `DOS-Hulk Attack` and `DoS-SlowHTTPTest Attack` are the same or similar to those in Table IV.

[8]https://pytorch.org
[9]https://scikit-learn.org/
[10]https://www.numpy.org/
[11]https://pandas.pydata.org/

The intrusion detection performance of each classification model is illustrated in terms of true-positive rate (Fig 2), false-positive rate (Fig 3) and time expense (Fig 4). The *true-positive rate* is the rate of the intrusion detected against all the intrusions happened, while *false-positive rate* is the rate of normal traffic been mistaken by the model as intrusions against all the normal traffics.

In Fig 2, x-axis denotes the true-positive rate, while the y-axis lists all types of intrusions recorded in the training dataset. The results of different machine learning classification models are denoted by different colours. As shown in Fig 2, different intrusion types illustrated different patterns in the traffic, some can be easily detected by all the classification models tested, such as `DoS-SlowHTTPTest`, `DoS-Hulk`, `DDoS-LOIC-UDP`, `DDoS-HOIC`, `FTP-BruteForce`. In other word, the traffic generated by these intrusions illustrated more distinct characters or patterns comparing to normal traffic. Some intrusions, however, shown more subtle differences. For example, for the intrusions `Infilteration`, `SQL Injection` and `BruteForce-Web`, only `the decision tree classification model` can provide a *true-positive rate* higher than 85%. To summarise, there is only one classification model perform well in all the intrusion types, the `decision tree classifier`, which is denoted by the light grey bars in Fig 2. Actually, as shown in Tables III – VIII, the accuracy of `decision tree classification` is straight 100%, except for `Infilteration`, which still performs better than the other models.

The *false-positive rate* is illustrated in Fig 3. The x-axis is the *false-positive rate* ranging from 0.00% to 20.00%, and the y-axis lists all types of intrusions in the training set. The *false-positive rate* is indicator of great significance in intrusion detection system, even more important than *true-positive rate*, for in real life, if an intrusion detection system generate too many false-positives, the alarms will not be taken seriously or even shut down by human users, which would cause greater danger than low *true-positive rate*. As shown in Fig 3, for most of the intrusions (more specifically, any intrusions other than `Infilteration`), *false-positive rate* generated by all the common machine learning classification models are as low as 2.00%, except for `Infilteration`, which experiences between $10.00\% \sim 18.50\%$.

The overhead in terms of time expense is illustrated in Fig 4 to demonstrate the efficiency of each machine learning classification models. The model that performs best in accuracy, `decision tree classification`, also cause less time than the peer classification models, consuming less than 20 seconds on all attack types.

In summary, as shown in the cross validation experiments in Section III-B, from the TPR, FPR, ROC, AUC and time cost point of view, although the MLP model perform higher accuracy amongst the other models, but it also consumes more time. Also as presented above in Section III-B, in the cross validation experiment, the `decision tree classification` fit best among the eight common classifier models, consuming less time. Thus, we claim the

TABLE III
MACHINE LEARNING RESULTS OF BOTNET ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00
	Bot	0.95	0.94	1.00	0.98	1.00	1.00	1.00	1.00
recall	Benign	0.94	0.93	1.00	0.98	1.00	1.00	1.00	1.00
	Bot	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00
f1-score	Benign	0.97	0.96	1.00	0.99	1.00	1.00	1.00	1.00
	Bot	0.97	0.97	1.00	0.99	1.00	1.00	1.00	1.00

TABLE IV
MACHINE LEARNING RESULTS OF FTP-BRUTEFORCE ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	FTP-BruteForce	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
recall	Benign	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	FTP-BruteForce	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
f1-score	Benign	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	FTP-BruteForce	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

TABLE V
MACHINE LEARNING RESULTS OF BRUTEFORCE-WEB ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BruteForce-Web	1.00	0.25	0.99	0.85	0.91	0.98	0.82	1.00
recall	Benign	1.00	0.80	1.00	0.99	0.99	1.00	0.99	1.00
	BruteForce-Web	0.45	1.00	0.99	0.99	0.98	1.00	0.99	0.99
f1-score	Benign	0.98	0.89	1.00	0.99	1.00	1.00	0.99	1.00
	BruteForce-Web	0.62	0.40	0.99	0.92	0.95	0.99	0.90	1.00

TABLE VI
MACHINE LEARNING RESULTS OF BRUTEFORCE-XSS ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00
	BruteForce-XSS	1.00	0.24	1.00	1.00	0.73	1.00	1.00	1.00
recall	Benign	1.00	0.93	1.00	1.00	0.99	1.00	1.00	1.00
	BruteForce-XSS	0.48	0.95	0.98	0.48	0.98	0.96	0.48	0.98
f1-score	Benign	0.99	0.97	1.00	0.99	1.00	1.00	0.99	1.00
	BruteForce-XSS	0.65	0.39	0.99	0.65	0.82	0.98	0.65	0.99

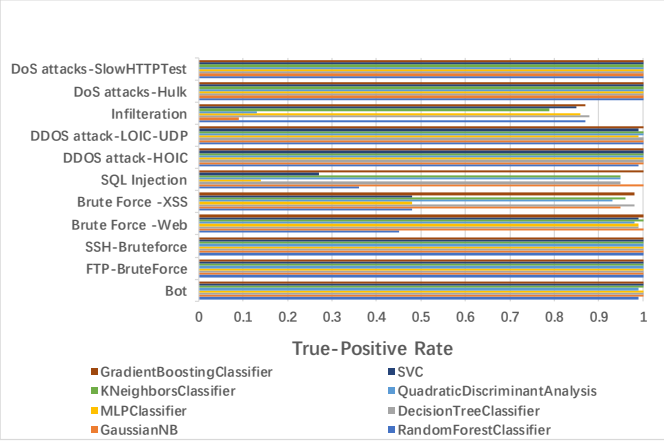


Fig. 2. The True-Positive rate for different classification models on different attack types.

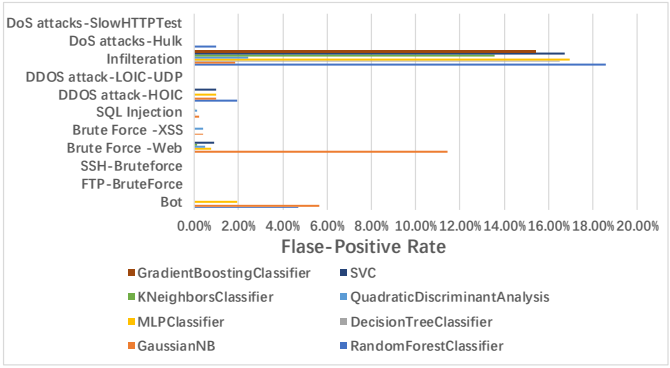


Fig. 3. The False-Positive rate for different classification models on different attack types.

decision tree classifier is most suitable model within all the eight common classifier models, with high accuracy, and reasonable time consuming, but if time overhead is not an issue, MLP performs the best.

C. Experiment 2: Zero-Day Intrusion Detection

In this subsection, we will present the detection results for the same 8 classifier models on Zero-day intrusions, that is intrusions in the test datasets have not appeared in the training datasets. We compose the test dataset by mixing data (both intrusion and benign data) from all eleven types of intrusions datasets. For the training datasets, we gradually increase the diversity of the intrusion types in the training datasets, from containing two types of intrusions to containing all eleven intrusions, and evaluate the intrusion detection performance on different diversities. The results are presented in Fig 5, 6, 7. We also use benign data collected from our production network and novel intrusions data (as shown in Table II, which are different from the eleven intrusions contained in CIC-AWS-2018 Dataset) from website and blogs of network security experts as test dataset, and use CIC-AWS-2018 Dataset as training data. The performance is evaluated and presented in

TABLE VII
MACHINE LEARNING RESULTS OF SQL INJECTION ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00
	SQL Injection	1.00	0.63	0.95	1.00	0.78	0.95	1.00	1.00
recall	Benign	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	SQL Injection	0.36	1.00	0.95	0.14	0.95	0.95	0.27	1.00
f1-score	Benign	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SQL Injection	0.53	0.77	0.95	0.24	0.86	0.95	0.43	1.00

TABLE VIII
MACHINE LEARNING RESULTS OF INFILTERATION ATTACK ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.82	0.51	0.85	0.83	0.52	0.79	0.82	0.85
	Infilteration	0.69	0.71	0.75	0.73	0.75	0.78	0.73	0.77
recall	Benign	0.62	0.96	0.71	0.68	0.96	0.77	0.69	0.73
	Infilteration	0.87	0.09	0.88	0.86	0.13	0.79	0.85	0.87
f1-score	Benign	0.70	0.67	0.77	0.75	0.68	0.78	0.75	0.79
	Infilteration	0.77	0.17	0.81	0.79	0.23	0.78	0.79	0.82

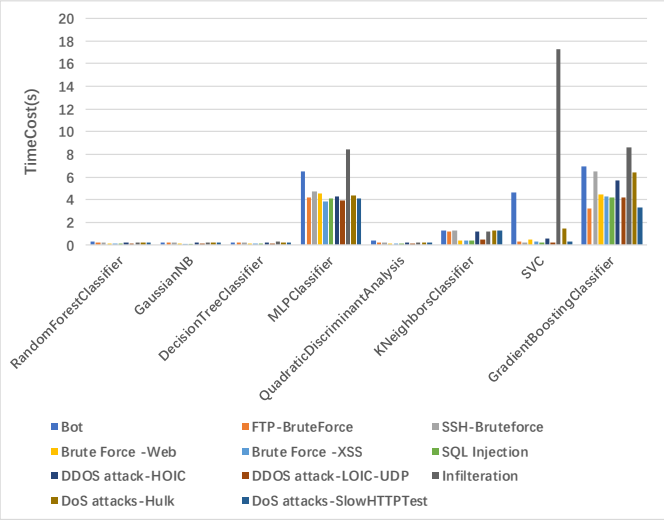


Fig. 4. The machine learning time consumption for different attack types.

Table IX, Fig 8, 9.

1) Use CIC-AWS-2018 Dataset as test dataset: : In Fig 5, x-axis denotes the true-positive rate, while the y-axis denotes the false-positive rate, and they form the receiver operating characteristic (ROC) curve. Different types of intrusion are shown in different colors. When the value of FPR is equal to the value of TPR, it indicates that the probability of correct or wrong prediction is equal, which is a random guess. At the same time, the value of the area under curve (AUC) is 0.5. Only if the value of TPR is greater than the value of FPR, the probability of correct prediction will be greater than the probability of wrong prediction. In other words, when the curve is closer to the upper left corner, the prediction result of the machine learning classification model is better. If the value of AUC is equal to 1, the machine learning classifier is a perfect classifier, which indicates that the accuracy of the prediction is 100%. As can be seen from Fig 5, when we adopt the One-Versus-One strategy, the effect of the eight classification models is very good, and the accuracy of most of them is above 95%.

When we adopt the One-Versus-One strategy, if the number of data types is N, we must design N(N-1)/2 classifiers. This method consumes more computer resources. Therefore, we will consider the One-Versus-Many strategy. Our goal is to detect intrusions with the accuracy in terms of *the true-positive rate* as high as possible and *the false-positive rate* as low as possible, especially when the intrusions has not been seen before (in other words **Zero-Day** attacks), we normalized all or part of the eleven intrusion types in train CIC-AWS-2018 Dataset and test datasets into one unique type "Evil". Thus there is an additional type of traffic in the datasets, "Evil". The goal is to transform this part of data into the "Evil" type through the training of the classifier model, and to detect whether the data in the data set has been invaded with the highest possible accuracy, especially **Zero-Day** attacks. As can be seen from Fig 6, when we change the label of an intrusion type into "Evil", only the ROC curve of MLP classifier in the data set of each intrusion type is always inclined to the upper left corner, and the AUC value is relatively large among all models, which is basically above 80%. As shown in Fig 7, with the number of intrusion categories in the "Evil" data increased, the effectiveness of the eight classification models gradually improved, such as the prediction accuracy of the decision tree model increased from 50% to 96%. Of course, the AUC value of the MLP classifier is as good as ever.

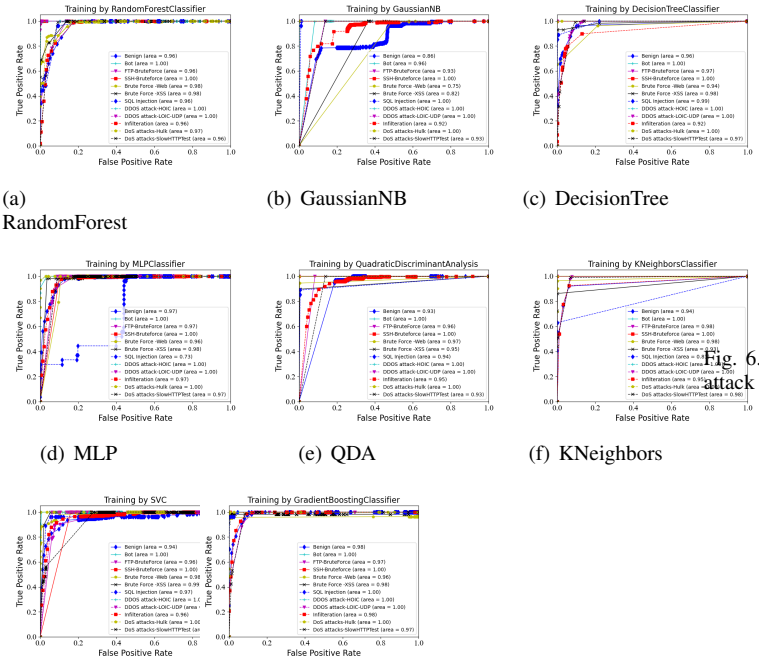
2) Use collected data as test dataset:: We go on test the models trained with the eleven intrusions from CIC-AWS-2018 Dataset with data collected from other sources. Benign data collected from our daily production network and novel intrusion data collected from website and blogs are mixed and used as test dataset. Results are presented in Table IX and Fig 8, 9.

As shown in Table IX, these popular machine learning models performs not as good as in Experiment 1 in detecting zero-day intrusions. `Naive Bayes` and `Quadratic Discriminate` get precision of 0.00 when detecting Benign data, and only 16% intrusion data. From Fig 8, the AUC of each machine learning model also deteriorates comparing to those in Experiment 1, where the intrusions are seen before in the training dataset. `Naive Bayes` and `Quadratic Discriminate` got 0.5 in AUC, which means these two model trained with CIC-AWS-2018 Dataset could not detect the unfamiliar intrusions at all. None of these models tested achieve above 0.9 in AUC, most of them (`Random Forest, Decision Tree, KNeighbour, SVC`) fall in the range

between 0.70 0.80. `MLP` and `Gradient Boosting` performs best, achieving almost 0.85.

Time overhead is illustrated in Fig 9. As shown in Fig 9, `SVM` is way slower than the other models, taking 2144 seconds, which is almost 100 2000 times of the other models. With the given datasets, `random forest`, `Quadratic Discriminate`, `Decision Tree`, `Naive Bayes` all takes less than 2 seconds to detect. `MLP` performs slightly worse, taking up to 34 seconds, `Gradient Boosting` takes more than 53 seconds, and `KNeighbour` need more than 100 seconds.

In summary, MLP performs the best, both when detecting known intrusions and unknown intrusions, decision tree performs well when detecting known intrusions, but its performance deteriorate rapidly when detecting unknown intrusions. The time overhead of MLP is also acceptable, considering its performance. Some of these famous models, `Gradient Boosting`, `KNeighbour`, `random forest` and `Quadratic Discriminate` do not perform well in detecting unknown intrusions, although all of them can detect known intrusions very well.

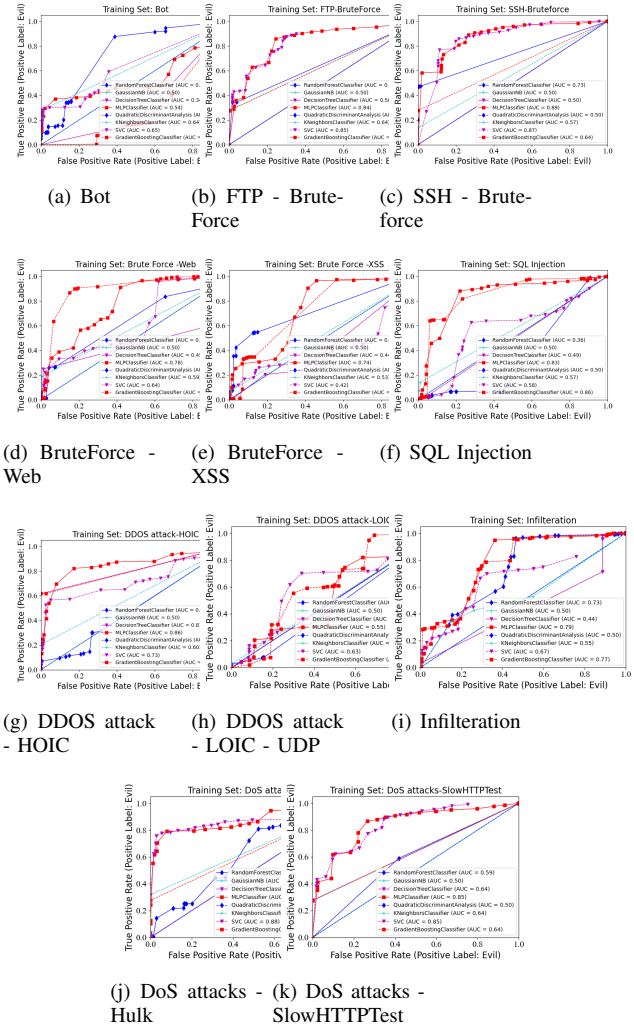


(a) RandomForest (b) GaussianNB (c) DecisionTree

(d) MLP (e) QDA (f) KNeighbors

(g) SVC (h) Gradientboost

Fig. 5. AUC - ROC curve for different attack types in Experiment 1 cross validation.



(a) Bot (b) FTP - Brute-Force (c) SSH - Brute-force

(d) BruteForce - Web (e) BruteForce - XSS (f) SQL Injection

(g) DDOS attack - HOIC (h) DDOS attack - LOIC - UDP (i) Infilteration

(j) DoS attacks - Hulk (k) DoS attacks - SlowHTTPTest

Fig. 6. AUC -ROC curve for different classification models on one certain attack types in Experiment 1 cross validation.

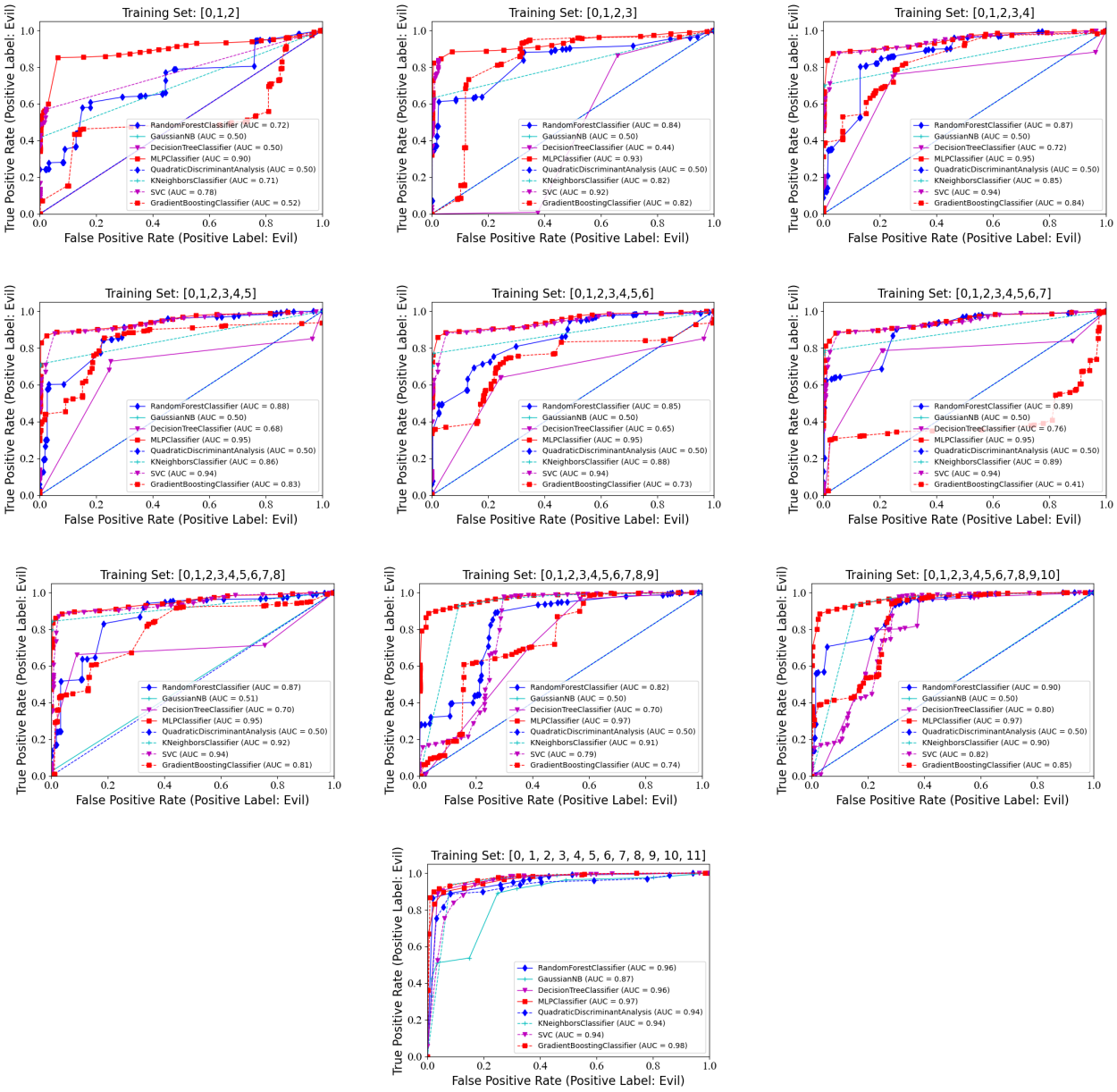


Fig. 7. AUC -ROC curve for different classification models on intrusion detections with gradually increased diversified training dataset in Experiment 2.

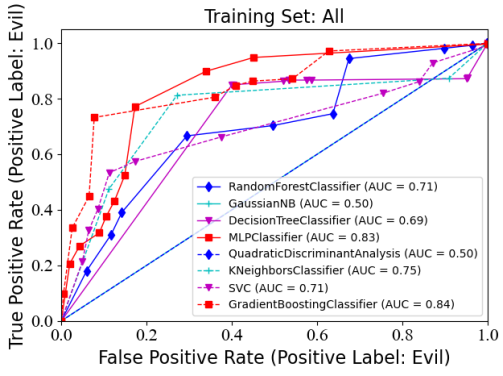


Fig. 8. The AUC-ROC curve for different classification models for zero-day intrusion detections using self-collected test dataset.

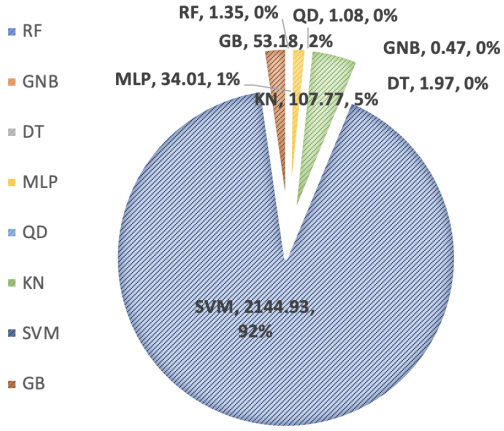


Fig. 9. The time overhead of different classification models for zero-day intrusion detections using self-collected test dataset, the time unit is second.

TABLE IX
DETECTION RESULTS OF ZERO-DAY DETECTION WITH COLLECTED NOVEL BENIGN AND INTRUSIONS DATASETS ON DIFFERENT CLASSIFICATION MODELS.

		Random Forest	Naive Bayes	Decision Tree	Neural Network (MLP)	Quadratic Discriminante	KNeighbors	Support vector classifier	Gradient boosting classifier
precision	Benign	0.94	0.00	0.95	0.76	0.00	0.95	0.88	0.95
	Evil	0.16	0.16	0.24	0.16	0.16	0.37	0.17	0.23
recall	Benign	0.02	0.00	0.47	0.01	0.00	0.73	0.14	0.42
	Evil	0.99	1.00	0.87	0.98	1.00	0.81	0.90	0.89
f1-score	Benign	0.05	0.00	0.63	0.02	0.00	0.83	0.24	0.59
	Evil	0.28	0.28	0.37	0.28	0.28	0.50	0.28	0.36

IV. CONCLUSIONS

In this paper, we take an intensive analysis on intrusion detection using the flow-based statistical data generated from network traffic packets with CICFlowMeter, tested on famous machine learning classification models. Eight common machine learning classifications models are tested on CIC-AWS-2018 Dataset and the datasets generated from real-life attacks and production networks. CIC-AWS-2018 Dataset which is collected by Amazon cluster networks, containing benign traffic and eleven different types of intrusions are used as training dataset, six different types of intrusions traffic data collected online and benign traffic data collected from our research production network are used as testing dataset. Cross validations over the training dataset are carried out on the eight common machine learning classification models, one model, decision tree classification, with the best performance on general adaptability, precision, and time consumption, is chosen to carry out the testing experiment. The testing results demonstrate that `MLP` performs the best, both when detecting known intrusions and unknown intrusions, `Decision Tree` performs well when detecting known intrusions, but its performance deteriorate rapidly when detecting unknown intrusions. The time overhead of `MLP` is also acceptable, considering its performance. Some of these famous models, `Gradient Boosting`, `KNeighbour`, `random forest` and `Quadratic Discriminate` do not perform well in detecting unknown intrusions, although all of them can detect known intrusions very well.

Much is left to be done in the future, for example, improve machine learning models that can detect more novel types of intrusions in real time, by collaborating with knowledge graph or other Artificial Intelligence technologies, like artificial immune system.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support from the National Natural Science Foundation of China (No.61991404, 61991400, 61973161, 61671244), State Key Laboratory of Synthetical Automation for Process Industries, the Fundamental Research Funds for the Central Universities, No.30921011103.

REFERENCES

[1] R. Hamilton, J. Iyengar, I. Swett, A. Wilk, et al., Quic: A udp-based secure and reliable transport for http/2, IETF, draft-tsvwg-quic-protocol-02.

[2] B. Mukherjee, L. Heberlein, K. Levitt, Network intrusion detection, IEEE Network 8 (3) (1994) 26–41.

[3] Y. Xiang, K. Li, W. Zhou, Low-rate ddos attacks detection and traceback by using new information metrics, IEEE Transactions on Information Forensics and Security 6 (2) (2011) 426–437.

[4] G. Chen, Y. Gong, P. Xiao, J. A. Chambers, Physical layer network security in the full-duplex relay system, IEEE Transactions on Information Forensics and Security 10 (3) (2015) 574–583.

[5] P. Garca-Teodoro, J. Daz-Verdejo, G. Maci-Fernndez, E. Vzquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers & Security 28 (1) (2009) 18–28.

[6] R. Sommer, V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, in: 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 305–316.

[7] L. Dhanabal, S. Shantharajah, A study on nsl-kdd dataset for intrusion detection system based on classification algorithms, International Journal of Advanced Research in Computer and Communication Engineering 4 (6) (2015) 446–452.

[8] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications Surveys & Tutorials 18 (2) (2016) 1153–1176.

[9] A. S. A. Aziz, E. Sanaa, A. E. Hassanien, Comparison of classification techniques applied for network intrusion detection and classification, Journal of Applied Logic 24 (2017) 109–118.

[10] J. Kim, J. Kim, H. L. T. Thu, H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 International Conference on Platform Technology and Service (PlatCon), IEEE, 2016, pp. 1–5.

[11] M. Ahmed, A. N. Mahmood, J. Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications 60 (2016) 19–31.

[12] S. Aljawarneh, M. Aldwairi, M. B. Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, Journal of Computational Science 25 (2018) 152–160.

[13] J. Undercofer, et al., Intrusion detection: Modeling system state to detect and classify aberrant behavior.

[14] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, An evaluation framework for intrusion detection dataset, in: 2016 International Conference on Information Science and Security (ICISS), IEEE, 2016, pp. 1–6.

[15] P. Aggarwal, S. K. Sharma, Analysis of kdd dataset attributes-class wise for intrusion detection, Procedia Computer Science 57 (2015) 842–851.

[16] I. Sharafaldin, A. Gharib, A. H. Lashkari, A. A. Ghorbani, Towards a reliable intrusion detection benchmark dataset, Software Networking 2018 (1) (2018) 177–200.
[17] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSP, 2018, pp. 108–116.
[18] B. J. Radford, B. D. Richardson, Sequence aggregation rules for anomaly detection in computer network traffic, arXiv preprint arXiv:1805.03735.
[19] I. Ullah, Q. H. Mahmoud, A two-level hybrid model for anomalous activity detection in iot networks, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2019, pp. 1–6.
[20] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, J. Ucles, Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification, in: Proc. IEEE Workshop on Information Assurance and Security, 2001, pp. 85–90.
[21] I. S. Thaseen, C. A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class svm, Journal of King Saud University-Computer and Information Sciences 29 (4) (2017) 462–472.

Qianru Zhou received her Ph.D degree in electrical engineering from Heriot-Watt University, Edinburgh, U.K. in 2018, and M.Sc. degree in optical engineering from Beijing University of Posts and Telecommunications, China, in 2013. She is currently an associate professor in Nanjing University of Science and Technology, Nanjing, China. Her research interests include autonomic network management, network security and network modeling.

Rongzhen Li received B.S. in Computer Science and Technology from Shandong University of Science and Technology, Qingdao, China, in 2017. He is currently working toward the Ph.D. degree with the School of Cyberspace Security, Nanjing University of Science and Technology, Nanjing, China. His research interests include vulnerability intrusion detection, machine learning and big data analysis.

Lei Xu receives his Bachelor, Master and PhD degrees in Communication and Information System at Nanjing University of Aeronautics and Astronautics, China, in 2006, 2009 and 2012, respectively. He is currently a professor at School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include 6G Wireless Network, Network Analysis, and Internet of Things.

Hongyi Zhu receives his Bachelor degree in Ocean engineering & technology at Zhejiang University, China in 2018. He is currently a postgraduate student at school of computer and engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include Machine Learning and Knowledge Graph.

Wanli Liu Wanli Liu is a professor at Nanjing Integrated Traditional Chinese and Western Medicine Hospital. His research interests include big data analysis.