A Comprehensive Analysis of Accuracies of Machine Learning Algorithms for Network Intrusion Detection

Anurag Das, Samuel A. Ajila, and Chung-Horng Lung(✉)

Department of Systems and Computer Engineering, Carleton University,
1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada
anuragdas@cmail.carleton.ca,
{ajila, chung}@sce.carleton.ca

Abstract. Intrusion and anomaly detection are particularly important in the time of increased vulnerability in computer networks and communication. Therefore, this research aims to detect network intrusion with the highest accuracy and fastest time. To achieve this, nine supervised machine learning algorithms were first applied to the UNSW-NB15 dataset for network anomaly detection. In addition, different attacks are investigated with different mitigation techniques that help determine the types of attacks. Once detection was done, the feature set was reduced according to existing research work to increase the speed of the model without compromising accuracy. Furthermore, seven supervised machine learning algorithms were also applied to the newly released BoT-IoT dataset with around three million network flows. The results show that the Random Forest is the best in terms of accuracy (97.9121%) and Naïve Bayes the fastest algorithm with 0.69 s for the UNSW-NB15 dataset. C4.5 is the most accurate one (87.66%), with all the features considered to identify the types of anomalies. For BoT-IoT, six of the seven algorithms have a close to 100% detection rate, except Naïve Bayes.

Keywords: Network intrusion detection · Supervised learning · UNSW-NB15 dataset · BoT-IoT dataset

1 Introduction

Due to the massive growth of computer networks and its many applications, the number of network flows has increased tremendously. The considerable large number of traffic flows leads to a massive amount of data, which eventually leads to the vulnerability of the data and network as a whole. One of the many challenges of cybersecurity research is to identify intrusion/anomaly in the traffic flow. A network Intrusion Detection System (IDS) is one of the solutions to detect such attacks before they compromise the network. An IDS monitors the normal traffic flows and identifies its characteristics or patterns. If a new flow does not follow the same characteristics, it might be an anomaly. Hence, an IDS may help identify even detect unknown attacks. Note that this paper uses intrusion and anomaly interchangeably.

Published by Springer Nature Switzerland AG 2020
S. Boumerdassi et al. (Eds.): MLN 2019, LNCS 12081, pp. 40–57, 2020.
https://doi.org/10.1007/978-3-030-45778-5_4

This research is an experimental investigation of nine machine learning algorithms on the dataset UNSW-NB15 (released in November 2015) [1] and seven machine algorithms on the recently released BoT-IoT dataset (released in November 2018) [2]. This research intends to discuss the following questions:

1. *Network Intrusion Detection:* How effective is it to detect network intrusion based on the traffic flow features present in datasets using different machine learning techniques?
2. *Types of Intrusion Classification:* Different cyberattacks can be stopped by different mitigation techniques. Hence classification of attack is as important as the detection of attacks. How effective can the classification of the types of attacks be from different features of network traffic flows present in datasets?
3. *Accuracy of models:* Which machine learning model has the highest accuracy for classifying the network anomalies for the selected datasets?
4. *Efficiency of models:* Which machine learning model is efficient for detecting network intrusion without compromising on accuracy? The earlier an attack is detected, the less harm it can generate on the network. Furthermore, by selecting a fewer number of features from the complete dataset, we can reduce the computation time a machine learning model takes to build.

The main contributions of the paper are: Firstly, comparing the accuracy and the time to build in the evaluation of network intrusion detection of the UNSW-NB15 dataset using nine machine learning techniques. Secondly, using the same nine machine learning techniques and nine different features selections, we compared and evaluated the performance of the various methods to identify the types of network intrusions in the UNSW-NB15 dataset. Thirdly, we analysed and evaluated the accuracy of and time to build seven machine learning techniques on the newly released BoT-IoT dataset. The premise upon which this research is based is to synthesis the previous research works on the UNSW-NB15 dataset [1–9]. Some (if not all) of related research works only used one or two machine learning algorithms to analyse the dataset, and in some cases do not even identify the different anomalies.

The rest of this paper is structured as follows. Section 2 presents background information about different supervised learning algorithms. Section 3 gives a literature review of previous research works and the different feature selection methods used in this paper. Section 4 presents the two datasets used in this research. Section 5 describes the methodology and the three sets of experiments. Section 6 provides the results and discussion and the conclusion is given in Sect. 7.

2 Background

2.1 Supervised Learning Algorithms

Machine learning is the study of algorithms that can learn complex relationships or patterns from empirical data and make accurate decisions [10]. Machine learning can be classified into supervised learning, semi-supervised learning, and unsupervised learning. Supervised learning deduces a functional relationship from training data that

generalizes well to the whole dataset. In contrast, unsupervised learning has no training dataset and the goal is to discover relationships between samples or reveal the latent variables behind the observations [11]. Semi-supervised learning falls between supervised and unsupervised learning by utilizing both labeled and unlabeled data during the training phase [10]. Among the three categories of machine learning, supervised learning is the best fit to solve the prediction problem in the auto-scaling area [11]. Therefore, this research focuses on supervised learning.

After conducting an in-depth search and review of research papers that have previously used the UNSW NB15 dataset, we selected nine machine learning algorithms that appear frequently in different papers [1, 3, 7], and [8].

Random Tree is an algorithm with a random number of features at each node, and it is used for classification [12]. This algorithm is very fast, but it suffers from overfitting. To overcome overfitting, Random Forest is used with this algorithm. We used the WEKA [13] implementation of this algorithm in which the Random Tree classifier constructs a tree that considers K random chosen attributes at each tree node. There is no pruning and has an option to estimate classifier probabilities (or target mean in the case of regression) based on a hold-out set (i.e. back fitting). We set the seed to be 1, that is, the random number seed used for selecting attributes.

Random Forest is an ensemble learning algorithm which can be applied on classification as well as a regression problem [12]. In this technique, lots of decision trees are produced at training time. For a regression problem, the mean is considered, and for the classification problem, the mode is used. Random Forest was designed to combat the overfitting problem in the random tree. Random Forest is a classifier for constructing a "forest" of random trees.

Bayesian Networks (WEKA Bayes Net) [13] - These networks show the probabilistic relations between different features with the target attribute (one which is to be classified) [12]. In this research, this algorithm is used to calculate the probability of different features with an impact on the anomaly. The dual nature of a Bayesian network makes learning a Bayesian network a two stage processes: first learn a network structure, then learn the probability tables. All Bayes network algorithms implemented in Weka assume that all variables are discrete finite and no instances have missing value [14]. In general, Bayes Network learning uses various search algorithms and quality measures [13]. In our case, we used the K2 search algorithm and the SimpleEstimator for the estimate function [13].

Naive Bayes - These are traditional classifiers and they are based on the Bayes theorem of independent relation between the features [12]. Although it is an old technique, this algorithm is still highly scalable and it can be used to build the fastest model for large dataset such as UNSW NB15. These classifiers are family of simple probabilistic classifiers with strong (naïve) independence. The assumption here is that the features of measurement are independent of each other.

k-Nearest Neighbours (k-NN) - k-NN is an algorithm which can be used for both regression and classification [12]. The model consists of training k closest samples in the feature space. In classification, the class having the maximum number of k nearest neighbours is chosen. Weights are assigned to nearer neighbours that contribute more to the result compared to the ones that are farther away. It is an instance-based learning algorithm where all the calculations are deferred until the final regression/classification.

Hence it is known as a lazy algorithm. This algorithm is called "IBk" in Weka [13]. It selects an appropriate value of k based on cross-validation and can also do distance weighting.

C4.5 - It is a decision tree-based classifier [12]. It is an extension of the classical ID3 algorithm from the same author - Ross Quinlan [15]. It uses information entropy and gain for decision making. On the Weka platform [13], it is called J.48, which is an implementation of C4.5 in Java. J.48 generates a pruned (or unpruned) C4.5 decision tree. The seed in this classifier is used for randomizing the data when reduced error pruning is used.

REPT - Reduced Error Pruning Tree (REPT) is a fast decision tree based on the C4.5 algorithm and it can produce classification (for discrete outcome) or regression trees (for continuous outcome). It builds a regression/decision tree using information gain/variance and prunes it using reduced-error pruning with back-fitting) [12, 13]. Missing values are replaced by breaking down the corresponding instances.

RIPPER - Repeated incremental pruning to produce error reduction (RIPPER) is an inductive rule-based classifier which is the optimized version of incremental reduced error pruning [12]. It uses incremental reduced-error pruning and a somewhat complicated global optimization step. It makes rules for all features. Depending on the satisfaction of those rules, a network flow is classified as normal or an anomaly. On the Weka platform, it is called Jrip [13]. Generally, the Weka Jrip implements a propositional rule learner.

PART (Partial Decision Tree) - Here rules are made according to the features of the past observations and classification of whether the data is an anomaly or normal is done according to the rules [12]. The Weka [13] implementation builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule.

These nine algorithms are either tree-based or partial tree or forest (a collection of trees) or networks (a form of tree). We have set the "seed" to 1 and the batch size to 100 where needed.

3 Literature Review

A number of research efforts [1–8, 12] have been conducted for network anomaly or intrusion detection using the UNSW-NB15 dataset. These approaches have certain limitations. Some research papers considered one or two machine learning algorithms. For instance, only the research works in [4] and [12] use more than one machine learning technique on the UNSW-NB15 dataset. Furthermore, the following research works: [4–6] and [12] do not identify the types of attacks. They only detect if a flow is normal or an anomaly. In addition, the research work in [12] does not adopt a feature selection method. Research works in [1–3] and [7, 8] do classify the attack types, but they only investigate a single machine learning technique.

In this paper, we investigate the detection, the types of attacks, and make a comparison between the effectiveness of nine different machine learning techniques as well as the impact of different feature selection techniques on those machine learning

algorithms. Our research, like [1] and [5], also does a benchmark of time taken for various machine learning techniques when applied together with specific feature selection methods.

The authors of [1] divided their network intrusion detection model into 3 stages. In the first stage, they applied Correlation-based feature selection on all the 45 features (as shown in Table 1) along with the genetic search. They used a statistical filter-based feature selection method on the complete dataset. Once they obtained the best features from stage 1, they applied a wrapper-based filter on those selected features only. The machine learning algorithm used in the wrapper-based filter was Random Forest. At the end of stage 2, the authors identified five best features, namely, *sbytes, tcprtt, synack, dmean and response_body_len*. In stage 3, they used the Random Forest classifier to detect the anomaly. They were able to improve the accuracy of the model from 94.70% to 99.63%. One problem in this approach is that only 5 out of 45 features were finally considered.

Table 1. Features for UNSW-NB15

Feature number	Feature name	Feature number	Feature name
1	id	23	dtcpb
2	dur	24	dwin
3	proto	25	tcprtt
4	service	26	synack
5	state	27	ackdat
6	spkts	28	Smean
7	dpkts	29	dmean
8	sbytes	30	trans_depth
9	dbytes	31	response_body_len
10	rate	32	ct_srv_src
11	sttl	33	ct_state_ttl
12	dttl	34	ct_dst_ltm
13	sload	35	ct_dst_dport_ltm
14	dload	36	ct_dst_sport_ltm
15	sloss	37	ct_dst_src_ltm
16	dloss	38	is_ftp_login
17	sinpkt	39	ct_ftp_cmd
18	dinpkt	40	ct_flw_http_mthd
19	sjit	41	ct_srv_ltm
20	djit	42	ct_srv_dst
21	swin	43	is_sms_ips_ports
22	stcpb	44	attack_cat
		45	label

The authors of [2] were the original authors of UNSW-NB15. Their method for feature selection has three parts: feature conversion, feature reduction and feature

normalization. After all the three steps, they selected the best features which are *Dttl, synack, swin, dwin, ct_state_ttl, ct_src_Itm, ct_srv_dst, Sttl, et_dst_sport_Itm,* and *Djit*. Association rule mining was also used to find features which are not correlated to each other, but highly correlated to the target attribute that the authors wanted to predict. They ranked all the 43 features (excluding id). From the ranking, they selected the top 25% (i.e., top 11 out of 43) features. Furthermore, Independent Component Analysis (ICA) was used to find the best features in [7].

The authors of [3] used the Weka tool [14] to select the optimal features. They used *CfsSubsetEval* (attribute evaluator) + *GreedyStepwise* method and *InfoGainAttibuteEval* (attribute evaluator) + *Ranker* method. The classifier Random Forest was used to evaluate the accuracy. The combination of features which generated the highest accuracy was selected. The five selected features are *service, sbytes, sttl, smean, ct_dst_sport_ltm*. Their test accuracy was around 83% for anomaly type classification.

The authors of [5] used a pure statistical filter-based subset evaluation (FBSE) method of correlation-based feature selection to detect Denial of Services (DoS) attacks. The final features selected are F7, F10, F11, F12, F18 and F32 (see Table 1). They used Artificial Neural Network to detect the attacks and obtained an accuracy of 81.34%. Their false alarm rate was quite high with 21.13%.

The authors of [6] trained a deep learning model on the entire dataset for 10-fold cross-validation. The most important features were then selected using the Gedeon method [9]. Gedeon method selects features which are unique from one another even if the information they provide is minor. They discarded the features which generate huge amount of redundant information. The accuracy obtained from the proposed model was 98.99% with a low false alarm rate of 0.56%.

The authors of [8] used Genetic Algorithm to find the best features. They used Support Vector Machine (SVM) to check the accuracy of the selected features.

4 Datasets

Two datasets have been selected for our experimental validation. They are UNSW-NB15 and BoT-IoT, which are described in the following subsections.

4.1 UNSW-NB15

This dataset was created by Moustafa and Slay [10]. The UNSW-NB15 dataset is a mixture of real normal traffic flow and synthetic attacks. The types of attacks and the number of each attack in the testing dataset are shown in Fig. 1. The testing dataset has 82,332 records (37,000 normal and 45,332 anomalies) and 45 attributes or features (see Table 1).

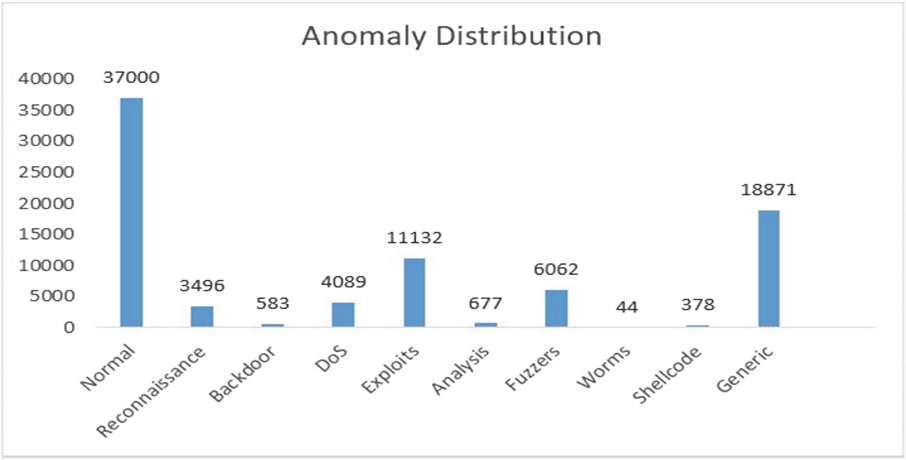

Fig. 1. Distribution of anomalies (attack types) in the UNSW-NB15 training dataset

4.2 The BoT-IoT Dataset

One of the original authors of UNSW-NB15 was also involved in creating the BoT-IoT dataset [11], as depicted in Fig. 2, in the Cyber Range Lab of UNSW Canberra Cyber Center. This dataset is a combination of standard and botnet traffic (hence the name). Attack distribution in the training dataset is depicted in Fig. 2. The training dataset has 2,934,817 records. The features used for the experiments were the top 10 features [11] selected by the creators of this dataset.

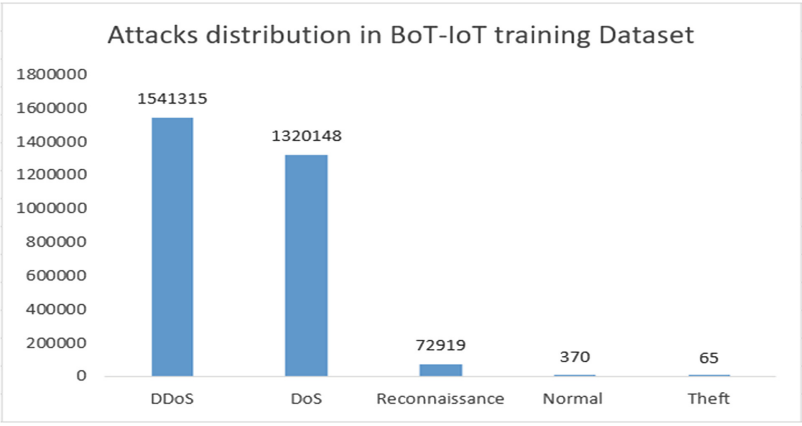

Fig. 2. Distribution of anomalies (attack types) in BoT-IoT training dataset

5 Methodology and Experiments

The specifications of the system environment for our experiments are shown in Table 2. Weka [14] tool was used for detection (training and testing), and the RStudio for data preprocessing with the R programming language. Microsoft Excel is used for data visualization. Three different experiment sets were conducted, which are described as follows.

Table 2. Hardware specifications

Processor	Intel(R) Xeon(R) @ 2.50 GHz (2 processors)
RAM	32.0 GB
Operating System	Windows 7 - 64 bit OS
Architecture	Microarchitecture - Ivy Bridge, Multiprocessor (2 Processors)

Experiment set 1: This set of experiments is to detect if a network flow is normal or an anomaly using supervised learning techniques. The nine machine learning algorithms described above have been evaluated and compared for accuracy (%) and the time taken to build the model (in seconds). The dataset used for this experiment set is the UNSW-NB15 training dataset.
Experiment set 2: In this set of experiments, the type of network attacks is also identified using the nine supervised learning algorithms for validation. Further, eight different feature selection techniques and the complete dataset (making a total of nine different feature sets) together with the nine different machine learning algorithms making 81 different combinations of feature selection methods and machine learning techniques to identify the types of attack. The dataset used is the UNSW-NB15 training dataset.
Experiment set 3: The types of network attacks are identified using seven supervised machine learning algorithms on the ten best features pre-selected by the authors of the BoT-IoT training dataset [11].

In addition, 10-fold cross-validation has been adopted [13] in our experiments. The standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified 10-fold cross-validation. The dataset is divided randomly into 10 parts in which the class is represented in approximately the same proportions as the full dataset. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. In the end, the average of all the iterations is calculated. As all the values have been tested at least once, this step helps in avoiding overfitting. Why 10? Previous extensive works in the domain have shown that the number 10 is about the right number of folds to get the best estimate of error.

5.1 Experiment Set 1 - Supervised Learning on UNSW-NB15 to Detect the Anomaly

Our first experiment-set used nine supervised learning algorithms on the UNSW-NB15 training dataset to detect the anomaly. The methodology in this experiment is that all the attributes (i.e. features) are considered and all nine machine learning algorithms are used to build anomaly detection models. The machine learning algorithms are Random Forest, Random Tree, Bayes Network, Naive Bayes, k-NN, C4.5, Reduced Error Pruning Tree, RIPPER and PART. The experimental results are presented in Table 3.

Table 3. Experiment I results

Machine learning algorithms	Accuracy (%)	False positive rate (%)	Precision (%)	Recall (%)	Time to build the model (s)
Random Forest	97.9121	2	97.9	97.9	57.25
Random Tree	96.1036	4	96.1	96.1	0.93
Bayes Network	81.6961	17.2	82.7	81.7	4.93
Naive Bayes	76.1952	21.4	79.1	76.2	0.69
k-nearest neighbours	93.4691	6.5	93.5	93.5	1.51
C4.5	97.3194	2.7	97.3	97.3	15.63
REPT	97.068	2.9	97.1	97.1	3.43
RIPPER	96.7582	3.2	96.8	96.8	185.36
PART	97.3109	2.7	97.3	97.3	53.69

As presented in Table 3, in terms of accuracy, Random Forest is the most accurate anomaly detection model with 97.91% closely followed by C4.5 (97.3194%), then PART (97.3109). Naïve Bayes is the least accurate considering the nine algorithms. Judging by the false positive rates and the precision (Table 3), there are little or no significant statistical differences between the following algorithms in terms of accuracy – Random Forest, Random Tree, C4.5, REPT, RIPPER, and PART. In terms of speed, that is, time to build, Naive Bayes is the fastest model with 0.69 s although the least accurate. Random Tree is equally fast with a build time of just 0.93 s and gives a high accuracy of 96.10%. Table 4 shows the confusion matrix of the Random Forest.

Table 4. Confusion matrix of the Random Forest algorithm

Classified as		
Normal	Anomaly	
36354	646	Normal
1073	44259	Anomaly

The diagonal in green indicates the correct classification. There are total of 37,000 normal flows and 45,332 anomalies in the UNSW-NB15 training dataset. Out of 37,000 normal, 36,354 were classified normal (98.25%) correctly, but 646 observations were classified anomaly incorrectly. Out of 45,332 anomalies, 1073 were classified normal incorrectly, but 44,259 were classified anomaly (97.63%) correctly.

5.2 Experiment Set 2 - Supervised Learning on UNSW-NB15 to Detect the Anomaly Type

Experiment set 2 was conducted to identify not only if a network traffic flow is normal or an anomaly, but also the types of anomaly. The objective is to support an appropriate action to mitigate it. The methodology for Experiment-set 2 is depicted in Fig. 3.

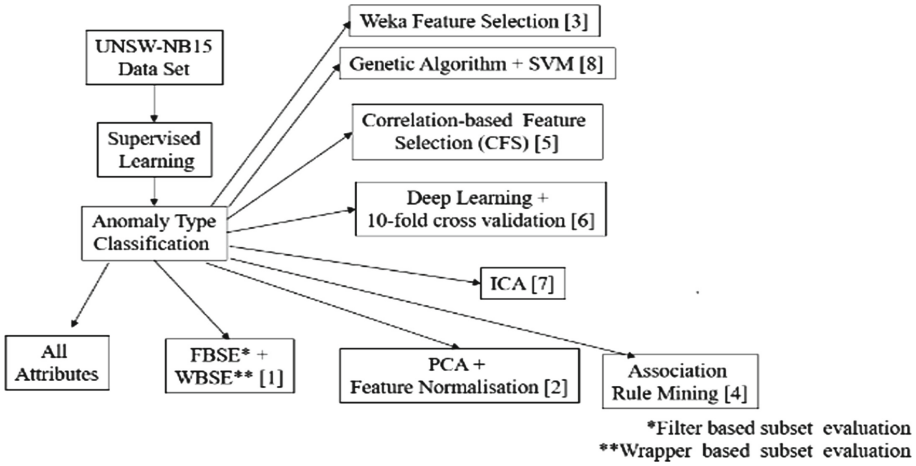

Fig. 3. Experiment 2 methodology

This experiment set 2 has nine sub-experiments with different feature sets. The experiments are based on methods previously published in the literature [1–8]. The features for the nine sub-experiments are described as follows

1. **All the features** of the UNSW-NB15 training dataset are considered.
2. **FBSE+WBSE -** Features selected by FBSE and wrapper WBSE. This experiment uses features suggested in [1].
3. **PCA + Feature Normalization -** Features selected by principal component analysis (PCA) and feature normalization are considered. This experiment is based on the features selected by [2].
4. **Weka Feature Selection -** Optimal features of UNSW-NB15 were selected using the Weka tool. This method was proposed in [3].
5. **Association Rule Mining -** Feature selection based on association rule mining. This experiment is based on work done in [4].

6. **Correlation based Feature Selection -** Pure statistical feature selection method based on correlation was used to select features. This experiment is based on [5], [16].
7. **Deep Learning with 10-fold CV -** The authors of [6] selected the best features which generated the highest accuracy for their deep learning model after 10-fold cross-validation [17–19]. These features were used in this sub-experiment.
8. **ICA -** Features used in this experiment were based on ICA proposed in [7].
9. **Genetic Algorithm with SVM -** Feature selection was done using the genetic algorithm. The classifier used in the genetic algorithm to check the highest accuracy was SVM. This methodology was proposed in [8].

Table 5. Accuracies of different algorithms for different feature sets for UNSW-NB15

Feature sets	Random Forest	Random Tree	Bayes Network	Naive Bayes	k-NN	C4.5	REPT	RIPPER	PART
All Features	87.08	84.17	65.28	46.16	80.62	**87.66**	86.62	80.24	87.05
FBSE + WBSE	82.85	80.92	74.25	17.95	76.59	82.56	82.33	76.67	82.30
PCA + Feature Normalization	85.85	83.48	71.55	41.87	81.27	85.78	85.13	79.58	85.65
Weka Feature Selection	82.99	82.85	74.55	57.57	82.5	83	82.8	76.44	82.9
Association Rule Mining	77.70	75.18	62.72	51.07	77.04	78.22	77.99	72.69	77.83
Correlation based Feature Selection	74.31	71.93	60.32	43.05	73.59	74.58	74.28	71.15	74.3
Deep Learning with 10-fold CV	85.57	83.8	66.55	56.59	84.17	86.16	85.10	79.24	85.91
ICA	85.68	83.59	74.27	37.87	80.12	85.31	84.86	77.6	85.03
Genetic Algorithm with SVM	77.35	74.18	69.68	35.83	71.67	76.05	76.80	71.86	75.78

Table 5 shows the experimental results. C4.5 generates the best accuracy for *All Features* (87.66%) followed by Random Forest (87.08%) then comes PART (87.05%) and the rest are REPT (86.62%), Random Tree (84.18%), k-NN (80.62%), Bayes Network (65.28%) and finally Naïve Bayes (46.16%)

Four algorithms (Random Forest, C4.5, REPT, and PART) are on par (roughly 83%) for FBSE + WBSE [2] and the rest are below 80% with Naïve Bayes as the worst (17.955%).

In the case of PCA + Feature Normalization [2], Random Forest produces the best accuracy (85.85%), followed by C4.5 (85.78%), PART (85.65%), REPT (85.13%), then Random Tree, k-NN, RIPPER and finally Bayes Network.

For Weka Feature Selection [3], six algorithms have roughly the same accuracies of around 83%. The maximum accuracy for Association Rule Mining [4] is 78.22% for C4.5 and the minimum is 51.07% for Naive Bayes.

The accuracy results for Correlation based Feature Selection [5] ranges from 74.58% for C4.5 to 43.05% for Naïve Bayes. Random Forest has the best accuracy for ICA [7].

All the accuracy measurements for the Genetic Algorithm with SVM [8] are below 80% with the highest as 77.35% for Random Forest and the minimum is 35.83% for Naïve Bayes.

In general, Random Forest came out to be the top in terms of accuracy for the feature sets for UNSW-NB15. This is closely followed by C4.5.

5.3 Experiment Set 3 - Supervised Learning on BoT-IoT to Detect the Anomaly Type

Bot-IoT training dataset [11] was used for experiment-set 3. It has around 3 million values. Only the top 10 features according to the original dataset authors were selected for this experiment set. Unfortunately, the Weka tool crashed for k-NN and RIPPER algorithms. It was unable to build models for these algorithms. This probably is due to the size of the dataset. Table 6 shows the results for the seven remaining algorithms and almost all the algorithms have around a 100% detection rate, except Naïve Bayes. These results are like that of the authors of [10]. Random Tree has high accuracy with a minimal build time of 96.98 s. Table 7 shows the confusion matrix of Random Tree for the BoT-IoT training dataset with 10-fold Cross-Validation.

Table 6. Accuracies of different algorithms for different feature sets for BoT-IoT

Algorithms	Accuracy %	False positive %	Precision %	Recall %	Time to build (seconds)
Random Forest	99.99	0	100	100	5628.96
Random Tree	**99.9937**	0	100	100	**96.98**
Naïve Bayes	**73.4121**	2.8	73.4	71.1	**11.46**
C4.5	99.99	0	100	100	448.57
REPT	99.99	0	100	100	205.96
Bayes Network	99.6	0.2	99.6	99.6	228.6
PART	99.99	0	100	100	1210.6

Table 7. Confusion matrix for Bot-IoT

Classified as					
Normal	DDoS	DoS	Reconnaissance	Theft	
349	1	6	12	2	Normal
2	1541262	48	3	0	DDoS
4	54	1320083	6	1	DoS
14	13	12	72878	2	Reconnaissance
1	0	0	3	61	Theft

The green diagonal shows the correct classification. As illustrated in Fig. 2, there are total of 370 normal flows, 1,541,315 DDoS, 1,320,148 DoS, 72,919 Reconnaissance, and 65 Theft in the BoT-IoT dataset. From the 370 normal flows, 349 (94.32%) were classified correctly. Out of 1,541,315 DDoS, 1,541,262 (99.99%) were classified correctly. For DoS 1,320,083 (99.9%) of 1,320,148 were classified correctly. 61 (93.85%) out of 65 theft were identified successfully.

6 Results and Discussions

In Experiment-set 1, nine machine learning algorithms have been evaluated and compared on UNSW-NB15 for network intrusion detection. In addition, the types of network intrusion are identified as well.

In Experiment-set 2, 81 (nine machine learning algorithms with eight different feature selection methods and all features together) different techniques have been compared. To the best of our knowledge, this is the first time all these nine methods are compared for the same dataset.

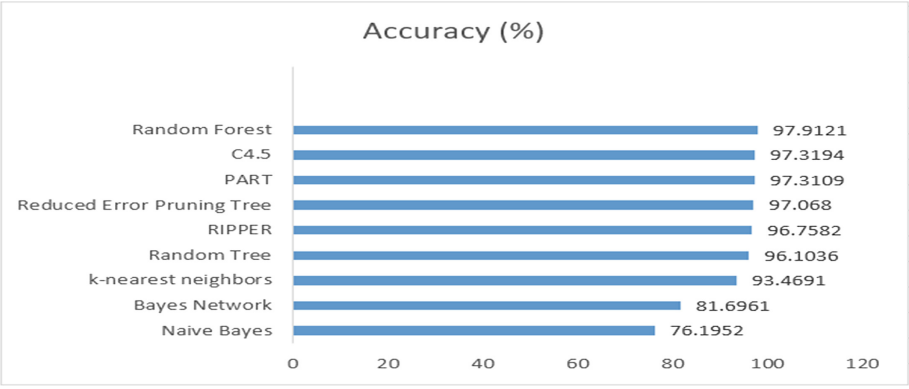

Fig. 4. Comparison of accuracy for different machine learning algorithms

Benchmarking has also been conducted on time taken to build each model [20–25].

Figure 4 displays the results of the comparison of different machine learning algorithms for anomaly detection on all the UNSW-NB15 dataset features. Random Forest provides the best accuracy (97.91%) but very costly in terms of computational time (57.25 s).

According to Fig. 5, Naive Bayes is the fastest algorithm for all the features. Random Tree is the most optimal algorithm with a high accuracy of 96.10% and second fastest with a build time of only 0.93 s.

Figure 6 is the comparison of all the different feature selection methods used from the existing literature that are applied to Random Forest to detect the anomaly.

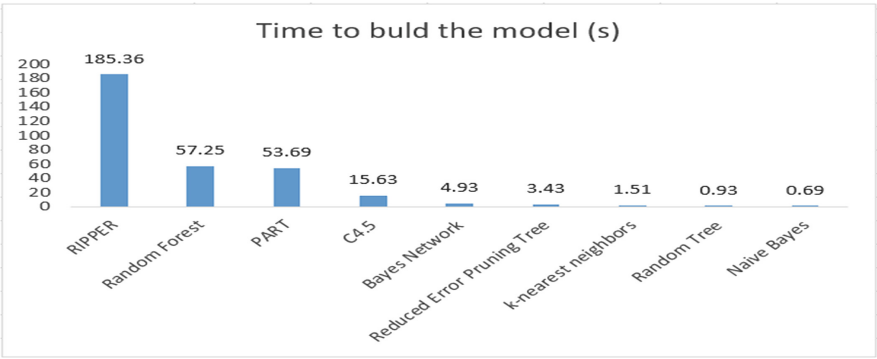

Fig. 5. Comparison of build time for different machine learning algorithms

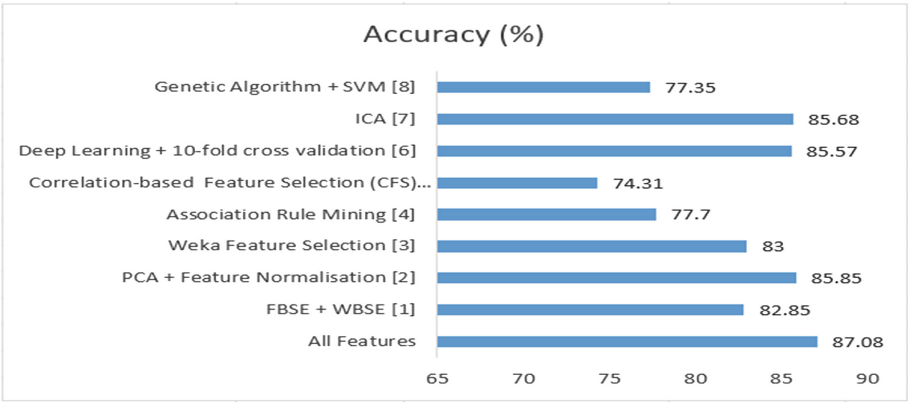

Fig. 6. Comparison of accuracies for feature selection methods on the Random Forest algorithm

When all features are used, Random Forest generates the best accuracy. For feature selection, PCA + Feature Normalisation, Deep Learning + 10-fold cross-validation and ICA all are valid regarding high accuracy.

According to Fig. 7, FBSE + WBSE, Weka feature selection is the fastest with around 19 s to build using the Random Forest algorithm. PCA + Feature Normalisation is a balanced approach for feature selection. It has the second highest accuracy with 85.85% and third fastest with 26.32 s.

According to Fig. 8, almost all the algorithms have around 100% detection accuracy for the BoT-IoT. The exception is Naïve Bayes with 73.4121%.

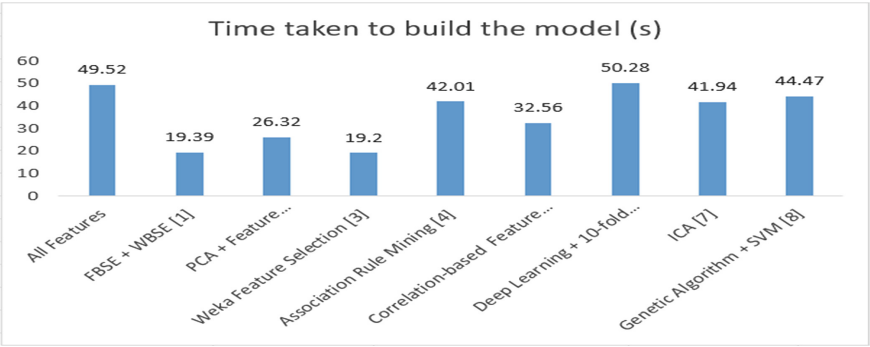

Fig. 7. Comparison of build time for feature selection methods on the Random Forest algorithm

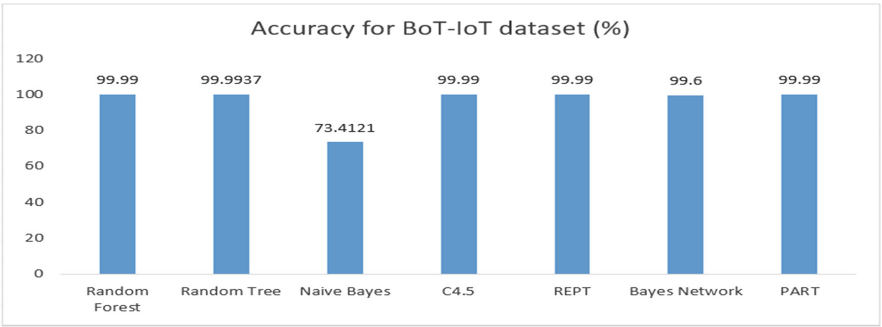

Fig. 8. Comparison of accuracies for different machine learning algorithms on BoT-IoT dataset

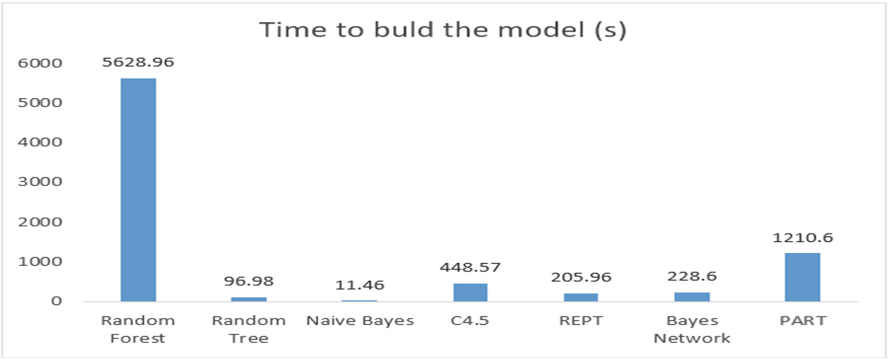

Fig. 9. Comparison of computational time for different machine learning algorithms on BoT-IoT dataset

Figure 9 shows that the time taken to build the model is very high for Random Forest. Naïve Bayes has the shortest time.

7 Conclusion and Future Work

We asked four questions at the beginning of this paper. The answers or the explanation to these questions are examined below.

Question 1 - *How effective is it to detect network intrusion based on the traffic flow features present in datasets using different machine learning techniques?*

As shown in Table 3 and Fig. 4, different machine learning algorithms give different accuracies for the UNSW-NB15 dataset. Firstly, this shows that the choice of a machine learning technique for anomaly detection is crucial based on the dataset. Secondly, there is a set of machine learning algorithms that have similar "accuracies" judging by the false positive rates and precisions which means that if speed is important one can choose a machine learning technique among others based on time to build the model.

Question 2 - H*ow effective can the classification of the types of attacks be from different features of network traffic flows present in datasets?* Table 5 shows the results of using the nine algorithms for nine different feature selection sets (81 different sub experiments). From this table, we can see that the set "All Features" shows "better accuracy" compared to other feature selections supposedly contains the best features. These results are somewhat different from previous research works (for example in papers [2] to [8]) that used feature selections. The only conclusion we can make here is that if timing (i.e., speed) is not important, then it may be better to use all the features if an efficient algorithm is used.

Question 3 - *Which machine learning model has the highest accuracy for classifying the network anomalies for the selected datasets?* To answer this question, there is no single machine learning algorithm that can be tagged as "the best" for classifying anomalies. However, based on the "nature" of the data and, these two datasets (UNSW-NB15 and BoT-IoT) the tree-based algorithms appear to perform better than non-tree based. In Table 3, other than k-NN; Bayes Network and Naïve Bayes are less accurate compared to the rest of the algorithms.

Question 4 - *Which machine learning model is efficient for detecting network intrusion without compromising on accuracy?* So far as this research is concerned and based on the dataset (i.e., UNSW-NB15), the best algorithm that does not compromise on accuracy is Random Tree (Table 3) with 96.9121% accuracy and a time of 0.93 s. The second is REPT with 97.3109% and 3.43 s. Other algorithms (Random Forest, PART, RIPPER, etc.) have higher accuracies but not that efficient in terms of build time.

In conclusion, in this research, supervised learning techniques were used to detect anomaly in the UNSW-NB15 dataset. Although Random Forest has the best accuracy (97.9121%) and Naïve Bayes is the fastest, Random Tree is the most optimal algorithm with an accuracy of 96.10% and the second fastest with a build time of only 0.93 s. Furthermore, supervised learning techniques were used to classify the types of attacks as well. C4.5 is the most accurate one (87.66%) with all the features considered. Eight different types of feature selection methods were used from existing literature to investigate the accuracy and timing of each model. PCA + Feature Normalisation [2] is a balanced approach for feature selection; it has the second highest accuracy with

85.85% and the third fastest with 26.32 s. Supervised Learning techniques were used on the BoT-IoT dataset to classify anomaly types. Random Tree is an optimal algorithm with almost perfect accuracy, and it is the second fastest one with just 96.98 s taken to build a model for 3 million network flows.

For future work, various feature selection methods can be applied to supervised learning algorithms. Weka can also be used for more feature selection methods. GPUs or distributed systems can be used to ease the computation burden. Unsupervised Learning algorithms can be applied to the BoT-IoT datasets. Deep learning models like Convolutional Neural Networks and Recurrent neural networks can be trained and compared to traditional machine learning algorithms regarding accuracy and speed.

References

1. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. In: IEEE Transactions on Big Data, p. 1 (2017)
2. Koroniotis, N., Moustafa, E., Sitnikova, B., Turnbull, B.: Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset (2018). https://arxiv.org/abs/1811.00701
3. Janarthanan, T., Zargari, S.: Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In: Proceedings of the IEEE 26th International Symposium on Industrial Electronics (2017)
4. Moustafa, N., Slay, J.: A hybrid feature selection for network intrusion detection systems: central points. In: Proceedings of the 16th Australian Information Warfare Conference, November 2015
5. Idhammad, M., Afdel, K., Belouch, M.: DoS detection method based on artificial neural networks. Int. J. Adv. Comput. Sci. Appl. **8**(4), 465–471 (2017)
6. Al-Zewairi, M., Almajali, S., Awajan, A.: Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system. In: Proceedings of the International Conference on New Trends in Computing Sciences, pp. 167–172 (2017)
7. Gharaee, H., Hosseinvand, H.: A new feature selection IDS based on genetic algorithm and SVM. In: Proceedings of the 8th International Symposium on Telecommunications, pp. 139–144 (2016)
8. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of the IEEE Military Communications and Information Systems Conference (MilCIS) (2015)
9. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J. **25**(1), 18–31 (2016)
10. Nikravesh, A.Y., Ajila, S.A., Lung, C.-H.: An autonomic prediction suite for cloud resource provisioning. J. Cloud Comput. **6**(3), 1–20 (2017). https://doi.org/10.1186/s13677-017-0073-4
11. Ajila, S.A., Bankole, A.A.: Using machine learning algorithms for cloud prediction models in a Web VM resource provisioning environment. Trans. Mach. Learn. Artif. Intell. **4**(1), 29–51 (2016)
12. Marsland, S.: Machine Learning: An Algorithmic Perspective, 2nd edn. Chapman and Hall/CRC, Boca Raton (2014)

13. Weka. https://www.cs.waikato.ac.nz/ml/weka/. Accessed June 2019
14. Bouckaert, R.R.: Bayesian Network Classifiers in Weka for Version 3-5-7, 12 May 2008. https://www.cs.waikato.ac.nz/~remco/weka.bn.pdf. Accessed June 2019
15. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986). https://doi.org/10.1007/BF00116251
16. Nguyen, H., Franke, K., Petrovic, S.: Improving effectiveness of intrusion detection by correlation feature selection. In: Proceedings of the International Conference on Availability, Reliability and Security, pp. 17–24 (2010)
17. Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell. **2**(1), 41–50 (2018)
18. Pervez, M.S., Farid, D.M.: Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: Proceedings of the International Conference on Software, Knowledge, Information Management and Applications, pp. 1–6 (2014)
19. Nguyen, H.T., Franke, K., Petrovic, S.: Towards a generic feature-selection measure for intrusion detection. In: Proceedings of the International Conference on Pattern Recognition, pp. 1529–1532 (2010)
20. Zainal, A., Maarof, M.A., Shamsuddin, S.M.: Feature selection using rough set in intrusion detection. In: TENCON IEEE Region 10 Conference, Hong Kong, pp. 1–4 (2006)
21. Muda, Z., Yassin, W., Sulaiman, M.N., Udzir, N.I.: Intrusion detection based on K-Means clustering and Naïve Bayes classification. In: Proceedings of the 7th International Conference on Information Technology in Asia, pp. 1–6 (2011)
22. Kumar, S., Yadav, A.: Increasing performance of intrusion detection system using neural network. In: Proceedings of the IEEE International Conference on Advanced Communications, Control and Computing Technologies, pp. 546–550 (2014)
23. Ingre, B., Yadav, A.: Performance analysis of NSL-KDD dataset using ANN. In: Proceedings of the International Conference on Processing and Communication Engineering Systems, Guntur, pp. 92–96 (2015)
24. Garg, T., Khurana, S.S.: Comparison of classification techniques for intrusion detection dataset using WEKA. In: Proceedings of the International Conference on Recent Advances and Innovations in Engineering, pp. 1–5 (2014)
25. Paulauskas, N., Auskalnis, J.: Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset. In: Proceedings of the Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1–5 (2017)