Performance Analysis of Network Intrusion Detection System using Machine Learning

Abdullah Alsaeedi[1], Mohammad Zubair Khan[2]
Department of Computer Science
College of Computer Science and Engineering
Taibah University, Madinah, KSA

***Abstract*—With the coming of the Internet and the increasing number of Internet users in recent years, the number of attacks has also increased. Protecting computers and networks is a hard task. An intrusion detection system is used to detect attacks and to protect computers and network systems from these attacks. This paper aimed to compare the performance of Random Forests, Decision Tree, Gaussian Naïve Bayes, and Support Vector Machines in detecting network attacks. An up-to-date dataset was chosen to compare the performance of these classifiers. The results of the conducted experiments demonstrate that both Random Forests and Decision Tree performed effectively in detecting attacks.**

***Keywords*—Intrusion Detection System (IDS); classifiers; AI; machine learning; KDD99; CICIDS2017; DoS; U2R; R2L**

I. Introduction

According to January 2019 statistics [1], the number of Internet users has surged compared to the previous year, with more than one million users using the web daily for the first time. There are 5.11 billion mobile users today, increased by 2% compared to the previous year. The number of Internet users worldwide is 4.39 billion, with an expansion of 366 million (9%) compared to January 2018. There are 3.48 billion social media users today, representing a 288 million (9%) increase since this time a year ago. In addition, 3.26 billion users utilized Internet-based social media on cell phones in January 2019. The development of 297 million new users represents a year-on-year increment in excess of 10%. Users are not simply Internet users but hackers too. A computer hacker is a skilled expert who uses technical expertise to hack computers. As per a report generated by AV-TEST [2], there are 350,000 malicious programs (malware) and unwanted applications every day. There are 1,250,000 hackers who make malware [2].

Today, political groups and businesses are progressively engaged in advanced digital warfare to combat harm to and intrusion on PC networks [3], as well as the theft of private content. It is important to ensure reliable measures to guard against the intrusion of powerful attackers over the network. These attacks fall into two categories [4]: passive and active. In passive attacks, the intruders obtain exchanged data through the network without causing any damage or disruption to communication or data. Examples of passive attacks include eavesdropping, non- participation, and monitoring. Active attacks are those that modify communication data or negatively affect operations. Instances of active attacks include jamming, message dropping, debasement, denial of service (DoS), and forging.

In recent years, the number of attacks has increased. Malware, botnets, spam, phishing, and DoS attacks have turned out to be consistent dangers for systems and hosts [5]. Therefore, efficient intrusion detection systems (IDS) have been designed and developed to detect these threats. Intrusion detection [6] is the process of observing and analyzing events in a computer system or network in order to detect possible incidents and to prevent any illegal access. The process commonly begins by automatically gathering information from various network sources and analyzing this information for potential security threats. As the number of threats and attacks are increasing day by day, a powerful IDS is necessary to secure the networks and computer systems. In this paper, detection and identification will be used interchangeably.

Available IDS are commonly categorized as either anomaly-based [7], signature-based [8], or a combination of both. The anomaly-based method focuses on identifying unfamiliar behaviors in a network by examining the network's activities. This method is effective in identifying attacks not encountered before, so it is effective on previously unseen attacks [9], [10]. On the other hand, the signature-based method uses a database that is built to identify attacks. It works by creating a database containing all traffic patterns associated with each detected attack. This strategy is very effective. However, it requires updating the databases continually to handle new data attacks and, regardless of whether the databases are up to date, they are defenseless against previously unseen attacks. Since these attacks are not in the database, they can't be counteracted.

A. Motivation and Objectives

The number of Internet users has increased in parallel with the increasing number of attacks made on the Internet daily. In recent years, dangers have increased in complexity as well, such as application attacks. These kinds of attacks are refreshed continuously. Hence, being able to develop and analyze the performance of the proposed IDS systems on newly released datasets such as the Intrusion Detection Evaluation Dataset (CICIDS 2017) containing up-to-date attacks is another motivation for this study.

As stated in the previous section, there are two fundamental methods to detect and distinguish attacks. These methods aim to guarantee data security and identify attacks based on either signature or anomaly. Developing an effective and reliable anomaly-based IDS to detect attacks properly with few false positives is a challenging task [11], [12]. For this reason, securing the networks and PCs against various kinds of attacks

has motivated us to develop a robust anomaly-based IDS by utilizing supervised machine learning classifiers.

The main objectives of this study are:

- to check the performance of machine learning algorithms (classifiers) that can be used to detect network anomalies or attacks.
- to validate the significance of results obtained using an IDS.

The rest of the paper is organized as follows. Section 2 describes the related background of anomaly types and datasets presented in the literature. Section 3 presents the methodology used for developing our IDS. Section 4 presents the experimental results and relevant discussion. Section 5 discusses the related works and Section 6 concludes the paper.

II. Background and Literature Survey

A. Anomaly Types and Network Intrusions

An anomaly is a sample of data that does not behave as well as normal samples [13]. There are three types of anomaly defined in the literature: point, contextual, and collective [14]. An anomaly can be considered as a point anomaly if it differs from the normal pattern of data samples in the entire dataset [14]. If a data sample behaves anomalously in a specific context or under specific conditions, it is referred to as a contextual anomaly. A set of similar instances that behave anomalously compared to other instances in the whole dataset is called a collective anomaly.

System security endeavors to shield the network system from attacks against the following three features: confidentiality, integrity, and availability [14], [15], [16]. The confidentiality feature is introduced to ensure that authorized users can only access data while these data are being transferred through networks. The integrity feature denotes that adding, modifying and deleting data can only be accomplished by the authorized user. The transmission data should maintain availability, which means that the services should always work promptly for the authentic user and that the network should be resilient against any kind of attack.

B. Types of Network Attacks

- Denial of Service (DoS) [14]: This attack is known as one of the most common kinds of intrusion and is intended to prevent legitimate users from accessing system resources or services. The DoS attacker may send a huge number of requests to a web server to prevent legitimate users from accessing services. DoS attacks can perform in two ways [17], [18]. The first way is bandwidth exhaustion, which aims to consume the bandwidth of the victims by flooding it with huge amounts of data. The second way is called resource consumption and its intention is to exhaust the victim's resources such as memory and processor. Many previous works [19], [20] have attempted to detect DoS attacks by utilizing artificial intelligence and machine learning approaches.
- Distributed DoS (DDoS): This attack is very similar to DoS in its intent, which revolves around preventing legitimate users from accessing the services. This kind of attack utilizes various computer systems as attack sources and attempts to flood the victim's devices including PC computers or IoT devices with inessential and useless requests. There are many IDSs [20], [21] proposed in the literature to detect DDoS attacks.
- Probing (information gathering) [14]: This aims to gather information about a machine, network structure, and network-connected devices. It focuses on collecting the security vulnerabilities of machines connected to the network. This kind of attack can be considered as the first step in other attacks.
- R2L/R2U (Remote to Local/Remote to User) [14], [22], [23]: In this attack, an attacker's intention is to gain access to the victim's PC to reveal system vulnerabilities, whereby an attacker attempts to get the privilege of sending packets over the Internet to get access to the system as a local user. The brute force method can be utilized to capture passwords and penetrate the system.
- U2R (User to Root) [14], [24]: In this kind of attack, the attacker focuses on gaining the privilege of administrator in order to access unauthorized files and manipulate important data [14]. The attacker may use system vulnerabilities to gain the privilege of administrator via sniffing passwords or a social engineering approach.
- Port Scan [25]: This aims to discover opened ports by scanning all ports in the victim's system and it can be considered as the initial phase of remote-to-local (R2L) attacks. The attackers use this kind of attack to discover more potential vulnerability that can help them in intruding on the victim's system. A number of IDSs have been proposed using machine learning approaches to detect this kind of attack.
- Botnet: In this kind of attack, the attackers use multiple devices connected to the Internet to get access to the victim's system by sending spams.
- Brute Force [26], [27]: A brute force attack is one of the most common attack types that threaten computer networks and break encryption. In this kind of attack, the attacker attempts to get user credentials by utilizing a repetitive method to guess username and password using automated software to get the valid account information of victims.
- Cross-site Scripting [28]: This is an attack that attempts to inject malicious codes in the client side of a website. It relies on weaknesses in the unencrypted websites and the information entered by users. The attackers may use dynamic content such as JavaScript and Flash to deliver malicious codes.
- SQL injection: This type of attack especially targets webservers, since SQL is a common language used in database servers. To get access to the information contained in the database server, attackers insert customized queries to obtain critical information such as personal information, passwords, or credit card

numbers for further malicious purposes. Also, such as in the case of webservers that contain a content management system (CMS), it is possible to insert the arbitrary code as a database register and execute it, thanks to a vulnerability in the CMS. Usually, most of the SQL injection attacks infect the vulnerable server. It may be possible for an attacker to go to a website's search box and type in a code that would force the site's SQL server to dump all its stored usernames and passwords for the site.

- Heartbleed [29]: This is an attack that can be considered as an impactful vulnerability in the OpenSSL that causes leaking of memory data. It allows attackers to remotely access sensitive data in a memory, including login credentials and private cryptographic keys.

C. Datasets

For evaluating the effectiveness of IDSs that are based on machine learning techniques, a huge amount of risky and riskless network traffic is required to train and test IDSs. Unfortunately, it is very difficult to use live network traffic publicly for security and privacy reasons. To deal with these issues, many datasets are available publicly to be used for testing and training IDSs. In this section, we discuss various datasets that are widely used to compare the performance of IDSs.

1) DARPA 98: DARPA dataset [30] was made by MIT Lincoln research to provide a complete benchmarking IDSs. In this dataset, the training and testing sets is split after simulating PC network of the United States Air Force's local. This dataset includes email and IRC messages, internet browsing, file transmission using FTP, Telnet activities. This dataset contains 38 kind of attacks that fall into the main four categories of attacks: Denial of Service (DoS), User to Remote (U2R), Probe, and Remote to Local (R2L).

2) KDD Cup 99: KDD Cup 99 dataset [31], [32] was made by the University of California for building and evaluating IDSs in the Third International Knowledge Discovery and Data Mining Tools Competition (The KDD Cup '99). It has been used commonly for evaluating anomaly-detection systems. The KDD Cup 99 dataset is split into training and testing sets. The training set comprises 4,898,431 and the test set comprises 311,029 records. The KDD Cup 99 contains 24 attack types in the training set and an additional 14 attack types in the testing set. These attacks fall into four main categories: Dos, R2L, U2R, and Probing. Compared to DARPA, KDD Cup 99 is commonly used for IDSs such as study presented in [33]. However, despite the fact that the KDD 99 dataset is a better option compared to DARPA 98, there are many redundancies in the KDD 99 dataset. These redundancies may affect the result of IDSs. Besides this, the size of the KDD 99 dataset is very large and minimizing the dataset may lead to losing some properties of the datasets.

3) NSL-KDD: In order to mitigate the weaknesses with the KDD dataset, Tavallaee et al. [24] built the NSL-KDD dataset to avoid redundancies by eliminating them. The dataset is not considered as representative of real networks. The NSL-KDD [34] dataset comprises 125,973 training samples and 22,544 testing samples. The size of the dataset is reasonable and experiments can be performed without any minimization [32].

4) ISCX 2012 : Although numerous datasets have been proposed for the identification of intrusions, these datasets are not up-to-date and do not reflect real-world data. To mitigate these problems, the Canadian Institute for Cybersecurity proposed a dataset named Intrusion Detection Evaluation Dataset [12], ISCX-IDS 2012, which was collected by monitoring seven-day network activity. The labeled dataset includes about 1,512,000 packets with 20 features.

The main features of this dataset are described in [35] and can be summarized as follows: real, normal, and malicious streams including FTP, HTTP, IMAP, POP3, SMTP and SSH protocols gathered as made, using real devices. All data are categorized and labelled. The collected datasets include various types of intrusion (Infiltrating, DoS, DDoS and Brute Force SSH).

5) CICIDS2017: The Canadian Institute for Cybersecurity at the University of New Brunswick made a new dataset for Intrusion Detection Evaluation named CICIDS 2017 [36], [37]. The CICIDS 2017 dataset comprises benign and the most up-to-date regular attacks, which takes after the true real-world data (PCAPs). Additionally, it incorporates the results of the network traffic analysis utilizing CICFlowMeter with named streams dependent on the time stamp, source and destination IPs, source and destination ports, conventions and attacks (CSV documents). This dataset contains a 5-day (July 3-7, 2017) data stream on a live network created by computers using up-to-date operating systems such as Windows Vista / 7 / 8.1 / 10, Mac, Ubuntu 12/16 and Kali.

This dataset has a few disadvantages. First, the size of the dataset is very large. Second, unlike KDD 99 and NSL-KDD datasets, there are no separate training and testing datasets. Finally, this is a new dataset so few studies have been used for building IDSs. In this paper, we decided to choose the CICIDS 2017 dataset for our experiment using Python, because it is an up-to-date dataset.

III. Related Work

In this section, different investigations utilizing machine learning to distinguish anomalies on PC networks have been analyzed sequentially. In each examination, the utilized AI algorithms, datasets and execution performance ratios are given. While choosing, these investigations have concentrated on the utilization of various machine learning algorithms and datasets.

Chebrolu et al. [38] applied a feature reduction approach to eliminate less informative attributes and used a Bayesian network (BN), Classification and Regression Trees (CART), and ensemble of both classifiers as intrusion detection systems. The Markov blanket (MB) model and decision tree (DS) were utilized to elect a subset of features. After that, the BN, CART, and the ensemble of both classifiers were examined. A hybrid approach of combining different feature selection approaches and ensemble classifiers utilized this approach on the KDD cup 99 intrusion detection dataset, and achieved different accuracies for each kind of attack: Normal (Benign): 100%, Probe: 100%, DOS: 100%, U2R: 84% and R2L: 84%.

Khan et al. [39] built an anomaly-detection system by combining a dynamically growing self-organizing tree (DGSOT) clustering algorithm with support vector machines (SVM). The intuition behind this combination is to aid SVM to cope with training datasets when they are very large. The random selection was applied to reduce the training data before, in this case, the SVM classifier attained accuracies of 98, 39, 23, 15, and 88% for the following attack kinds: Normal, Dos, U2R, R2L, and Probe, respectively. On the other hand, clear improvements were obtained after combining DGSOT with SVM for DOS, R2L, and prob attacks, achieving accuracies of 97, 43, 91%, respectively.

Yassin et al. [40] proposed an architecture that combines K-Means clustering and Naïve Bayes classifier to increase the detection rate and decrease both false positive and false negatives. They computed the detection rate (precision) and the false alarm (false positive) rate to measure the performance of IDS. The proposed architecture was evaluated using the ISCX 2012 Intrusion Detection Evaluation Dataset. A high accuracy of 0.99 and high detection rate of 98.8 were obtained on the testing data. Besides, a false positive rate of 0.13% was obtained.

Gaikwad et al. [41] proposed a new IDS using a bagging ensemble method with REPTree as the base estimator. Applicable features from the NSL-KDD dataset were selected manually to improve classification accuracy and decrease the false positive rate. The performance of the proposed bagging ensemble method was assessed using classification accuracy, model building time, and false positive rates. The results of the experiments conducted demonstrated that the bagging ensemble method with REPTree as base estimator attained a classification accuracy of 99.67 on 10-fold cross validation and 81.29% on the test dataset.

Divyasree and Sherly [42] proposed an effective IDS utilizing ensemble core vector machine (CVM) approach. CVMs are algorithms which work based on the idea of Minimum Enclosing Ball. The proposed IDS was built to detect attacks such as U2R, R2L, Probe and DoS attacks. The KDD Cup 99 dataset was used to train and test the classifiers. The chi-square test was used to elect the relevant features for each kind of attack to reduce the dimensionality of features. The experimental results demonstrated that CVM models achieved high accuracy for each kind of attack. The CVMs attained accuracies of 0.99, 0.945, 0.76, and 0.937 for Dos, Probe, R2L, and U2R, respectively.

Akram Boukhamla et al. [22] built a new dataset called CICIDS2017 to compare the effectiveness of IDSs. They used principal component analysis (PCA) to reduce the dimensionality of the features. The preprocessing phase included removing missing, redundant or infinite values and removing all nominal feature such as flow ID, source IP, destination IP, timestamp. The minimized CICIDS 2017 dataset was assessed using KNN, C4.5 and naïve Bayes classifiers. The outcomes of their experiment showed that NB attained the highest detection rate (recall) for DDoS, XSS, SqlInjection, and Infiltration attacks, while KNN attained a higher detection rate for PortScan and Botnet attacks. The C4.5 classifier achieved the highest detection rate for Brute Force attacks.

Dong Seong Kim et al. [43] proposed combining Genetic Algorithm (GA) with SVM to improve the performance of the SVM-based IDS. The proposed system in [43] showed the novelty of using GA for selecting optimal features and for choosing optimal parameters for the SVM classifiers. The outcomes of their experiments proved that the system could achieve a detection rate of 0.99 on the KDD 1999 dataset, considering the best performing IDS compared to the traditional SVM.

Ashraf et al. [23] compared the performance of Naive Bayes, J48, and Random Forest (RF) on 20% the NSL-KDD dataset. The valuable attributes were selected using the filter method in WEKA where Info gain was used as attribute evaluator. The collected results proved that RF performed better than NB and J48. Accuracies of 96.27, 99.17, and 99.71 were attained by NB, J48, and RF, respectively. The RF achieved an F-Measure score of 0.997, which is the highest score compared to NB and J48.

Kumar et al. [44] proposed an effective IDS based on a modified NB classifier to overcome the drawback with the traditional NB at detecting intrusions. They compared the performance of modified NB with Naïve Bayes, J48, and REPTree on the NSL-KDD dataset. The modified NB attained the highest accuracy of 92.34. Besides, the performance of the modified NB, traditional NB, J48, and REPTree were measured based on several feature selection methods including Correlation-based, Information Gain, and Gain Ratio. The modified NB attained an accuracy of 98.94 when gain-ratio was used for feature selection.

IV. Methodology

For the experiments, the well-known CICIDS 2017 dataset [36] was chosen. The reason behind selecting this dataset is that it is among the most up-to-date datasets containing the most up-to-date attacks. Table I reports the datasets used in the experiments along with the statistics. Decision Tree (DS), Gaussian Naïve Bayes (GNB), Random Forest (RF), and Linear Support Vector Machines (SVM) were selected as classifiers. The experiments were conducted in a Python environment with the scikit-learn. The classifiers' performances in this study were measured using classification accuracy, precision, recall, and F-score. It is important to highlight that these metrics were computed using the weighted average. The intuition behind selecting the weighted average was to calculate metrics for each class label and take the label imbalance into the account. The performance of classifiers was evaluated based on 5-fold cross-validation to split the datasets into five consecutive folds, one of them for testing and the remaining folds for training.

The following algorithm shows the steps used for the experiments. A list of datasets and a list of classifiers were provided first and then proceeded to iterate over all datasets, as shown in Line 7. The datasets were split into training and testing sets based on 5-fold cross-validation with shuffling of the data before splitting, as shown in Line 8. The loop in Lines 9–20 focused on training the classifiers, obtaining predictions, and computing evaluation metrics for each fold. The average scores were computed since the datasets were split using 5-folds. The process from Lines 7–28 was iterated through all provided datasets.

Input : *Datasets, Classifiers*
Result: *AvgAccurecy, AvgRecall, AvgPrecision, and AvgF-score*

Datasets ← {*DS1*, *DS2*, *DS3*, *DS4*, *DS5*, *DS6*, *DS7*};
Classifiers ← {*RF*, *DS*, *SVM*, *GNB*};
AllAccuracyScores ← {};
AllRecallScores ← {};
AllPrecisionScores ← {};
AllFScores ← {};
for DS ∈ $Datasets$ **do**
  for $Xtrain, Xtest$ ∈ KFold (nsplits = 5, shuffle = $True$).$split(DS)$ **do**
    for clf ∈ $Classifiers$ **do**
      clf ← *TrainClassifier* ($clf, Xtrain, XtrainLabels$);
      predictions ← *predict* ($cls, Xtest$);
      Accurecy ← *ComputeAccurecy* ($predictions, XtestLabels$);
      Recall ← *ComputeRecall* ($predictions, XtestLabels$);
      Precision ← *ComputePrecision* ($predictions, XtestLabels$);
      F-score ← *ComputeFmeasure* ($predictions, XtestLabels$);
      AllAccuracyScores ← *AllAccuracyScores* ∪ (*clf*, *Accurecy*);
      AllRecallScores ← *AllRecallScores* ∪ (*clf*, *Recall*);
      AllPrecisionScores ← *AllPrecisionScores* ∪ (*clf*, *Precisionll*);
      AllFScores ← *AllFscoreScores* ∪ (*clf*, *F-score*);
    end
  end
  for clf ∈ $Classifiers$ **do**
    AvgAccurecy ← *ComputeAvgAccurecy* ($AllAccuracyScores.get(clf)$);
    AvgRecall ← *ComputeAvgRecall* ($AllRecallScores.get(clf)$);
    AvgPrecision ← *ComputeAvgPrecision* ($AllPrecisionScores.get(clf)$);
    AvgF-score ← *ComputeAvgFmeasure* ($AllFScores.get(clf)$);
  end
end

Algorithm 1: The experimental procedure for IDS using supervised machine learning algorithms

V. Results

Table I summarizes the average scores achieved by classifiers for the datasets. It is clear that DS and RF attained the highest scores compared to other classifiers. Among these classifiers, the CNB classifier performed badly in all cases.

TABLE I. The average scores obtained for all classifiers

Datasets	Classifier	Accuracy	Precision	Recall	F-score
Dataset1	DS	0.99987	0.999867	0.999867	0.999867
	GNB	0.80790	0.855282	0.807899	0.79449
	RF	0.99996	0.999956	0.999956	0.999956
	SVM	0.96272	0.963079	0.962718	0.962772
Dataset2	DS	0.999895	0.999895	0.999895	0.999895
	GNB	0.690785	0.792843	0.690785	0.64342
	RF	0.99993	0.99993	0.99993	0.99993
	SVM	0.960712	0.960709	0.960712	0.960711
Dataset3	DS	0.999948	0.999948	0.999948	0.999948
	GNB	0.353517	0.989691	0.353517	0.510318
	RF	0.999869	0.999869	0.999869	0.999869
	SVM	0.959693	0.986486	0.959693	0.971018
Dataset4	DS	0.999983	0.999983	0.999983	0.999981
	GNB	0.972659	0.999844	0.972659	0.986003
	RF	0.999965	0.999965	0.999965	0.999957
	SVM	0.997226	0.999741	0.997226	0.998525
Dataset5	DS	0.997709	0.999989	0.997709	0.997741
	GNB	0.773101	0.970593	0.773101	0.865443
	RF	0.997533	0.999978	0.997533	0.997448
	SVM	0.986342	0.982202	0.986342	0.988383
Dataset6	DS	0.999989	0.99778	0.999989	0.999989
	GNB	0.53325	0.988128	0.53325	0.665683
	RF	0.999978	0.997449	0.999978	0.999978
	SVM	0.982744	0.990504	0.982744	0.982443
Dataset7	DS	0.999523	0.999523	0.999523	0.999523
	GNB	0.498051	0.817196	0.498051	0.497581
	RF	0.999429	0.999429	0.999429	0.999429
	SVM	0.833831	0.854528	0.833831	0.834621

Fig. 1 shows the barplots of the mean of accuracy scores attained by the classifiers. It is obvious that DS and RF classifiers attained very similar accuracies. The scores of the classification accuracy showed clear outperforming by DS and RF classifiers. The mean value of accuracy scores attained by DS and RF was 0.999, which is better than any other learners. The worst accuracies were attained by GNB for all datasets.

Table II shows the p-values obtained using the paired t-test

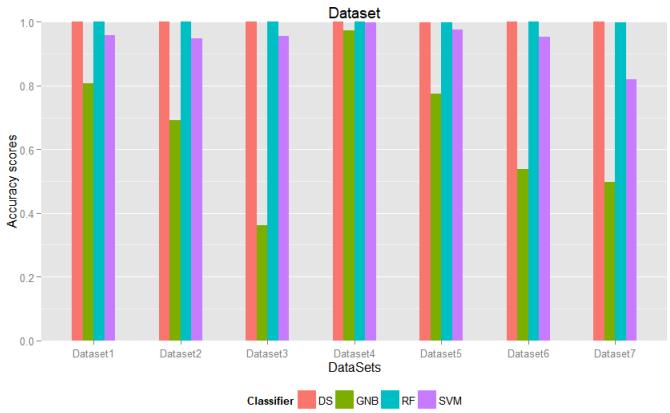

Fig. 1. Accuracy scores obtained by the classifiers

after comparing the accuracy scores attained by possible pairs of classifiers. By comparing the DS and RF, the p-values are larger than 0.05, accepting the null hypothesis that the mean difference between accuracies of both classifier is the same. For the comparison of DS and GNB, the p-values are less than 0.05, rejecting the null hypothesis.

TABLE II. P-values of accuracy scores

	DS vs. RF	DS vs. GNB	DS vs. SVM	RF vs. GNB	RF vs. SVM	GNB vs. SVM
DS1	0.0919	1.321e-11	0.005925	1.691e-11	0.005817	4.66e-05
DS2	0.0919	1.321e-11	0.005925	1.691e-11	0.005817	4.66e-05
DS3	0.01893	8.975e-08	0.004048	8.947e-08	0.004102	1.46e-06
DS4	0.189	5.311e-05	0.05847	5.428e-05	0.05899	0.0001974
DS5	0.829	4.641e-09	0.1661	6.123e-09	0.1655	8.919e-05
DS6	0.7396	1.603e-07	0.1965	1.605e-07	0.1967	0.0001363
DS7	0.02938	6.946e-09	0.005506	7.124e-09	0.005517	0.0005039

Fig. 2 illustrates the barplots of the mean of the precision scores obtained by the four classifiers. It is apparent that both DS and RF classifiers attained the highest precision scores. For all datasets, the mean precision scores attained by DS and RF are about 0.99. For dataset4, it is clear that all classifiers obtained a mean precision value of 0.99 for all classifiers.

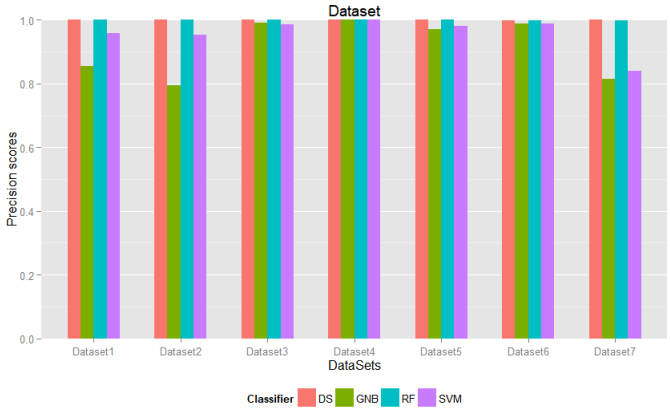

Fig. 2. Precision scores obtained by the classifiers

Table III shows the p-values obtained using the paired t-test after comparing the precision scores achieved by possible

pairs of classifiers. By comparing the DS and RF, the p-values are larger than 0.05 in the majority of cases, accepting the null hypothesis that the precision values of both classifiers are the same. The p-values are less than 0.05 after comparing the precision scores attained by DS and GNB, rejecting the null hypothesis that the precision values of DS and GNB are equal.

TABLE III. P-VALUES OF PRECISION SCORES

	DS vs. RF	DS vs. GNB	DS vs. SVM	RF vs. GNB	RF vs. SVM	GNB vs. SVM
DS1	0.0918	2.209e-11	0.006157	2.674e-11	0.006045	0.0002085
DS2	0.07511	2.152e-09	0.02233	2.133e-09	0.02235	0.0002795
DS3	0.0188	4.758e-07	4.674e-05	6.244e-07	4.926e-05	0.006344
DS4	0.1908	0.005962	0.008912	0.002459	0.008954	0.02417
DS5	0.7395	3.091e-08	0.002439	3.144e-08	0.002472	0.03249
DS6	0.3871	2.084e-07	0.01786	3.7e-07	0.01919	0.9129
DS7	0.02711	1.949e-07	0.004181	1.889e-07	0.004191	0.4181

Fig. 3 depicts the barplots of the recall scores obtained by the four classifiers. It is noticeable that the GNB classifier performed badly. It is clear that DS and RF attained the highest recall scores, compared to GNB and SVM. The mean values of recall attained by GNB were 0.807, 0.69, 0.36, 0.97, 0.77, 0.53, and 0.49 for Dataset 1 to Dataset 7, respectively.

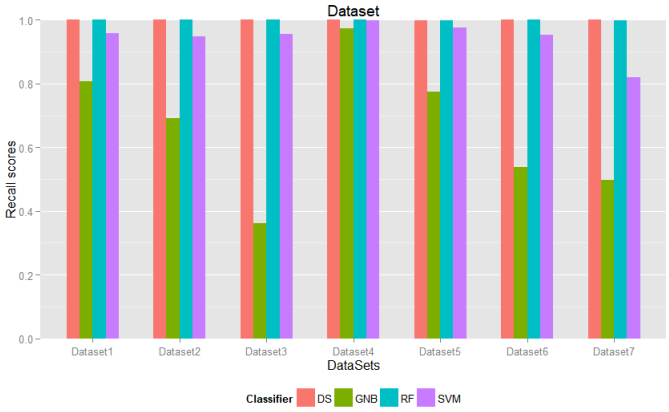

Fig. 3. Recall scores obtained by the classifiers

Table IV shows the p-values obtained using the paired t-test after comparing the accuracy scores attained by possible pairs of classifiers. By comparing the DS and RF, the p-values are larger than 0.05, accepting the null hypothesis that the mean values of both classifiers are the same. For the comparison of DS and GNB, the p-values are less than 0.05, rejecting the null hypothesis.

TABLE IV. P-VALUES OF RECALL SCORES

	DS vs. RF	DS vs. GNB	DS vs. SVM	RF vs. GNB	RF vs. SVM	GNB vs. SVM
DS1	0.0919	1.321e-11	0.005925	1.691e-11	0.005817	4.66e-05
DS2	0.07513	9.111e-09	0.03011	9.145e-09	0.03012	9.354e-05
DS3	0.01893	8.975e-08	0.004048	8.947e-08	0.004102	1.46e-06
DS4	0.189	5.311e-05	0.05847	5.428e-05	0.05899	0.0001974
DS5	0.829	4.641e-09	0.1661	6.123e-09	0.1655	8.919e-05
DS6	0.7396	1.603e-07	0.1965	1.605e-07	0.1967	0.0001363
DS7	0.02938	6.946e-09	0.005506	7.124e-09	0.005517	0.0005039

Fig. 4 illustrates the barplots of the F-scores obtained by the four classifiers. It is clear that the GNB classifier performed badly. The mean values of F-scores attained by GNB were 0.79, 0.646, 0.518, 0.985, 0.866, 0.67, and 0.498 for Dataset 1 to Dataset 7, respectively. Besides, for Dataset 1 to Dataset 7, the mean values of F-scores achieved by SVM were 0.957, 0.948, 0.969, 0.998, 0.98, 0.96, and 0.80, respectively.

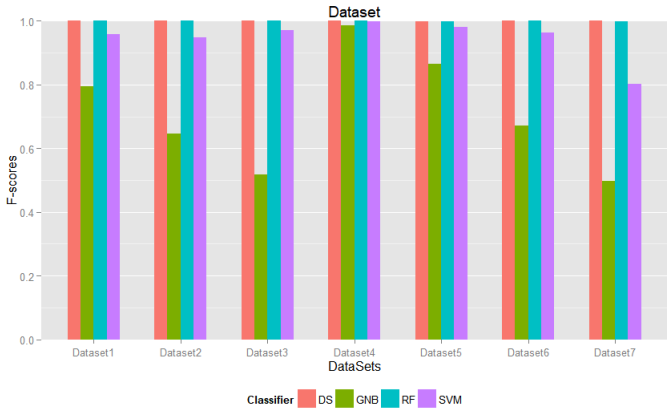

Fig. 4. F-scores obtained by the classifiers

Table V shows the p-values obtained using the paired t-test after comparing the F-scores attained by possible pairs of classifiers. By comparing the DS and RF, the p-values are larger than 0.05, accepting the null hypothesis that the F-scores of both classifiers are the same. For the comparison of DS and GNB, the p-values are less than 0.05, rejecting the null hypothesis, indicating that there is a significant difference between the F-scores of DS and GNB.

TABLE V. P-VALUES OF F-SCORES

	DS vs. RF	DS vs. GNB	DS vs. SVM	RF vs. GNB	RF vs. SVM	GNB vs. SVM
DS1	0.0919	2.437e-11	0.005924	3.133e-11	0.005816	3.453e-05
DS2	0.07513	2.492e-08	0.03122	2.499e-08	0.03123	5.904e-05
DS3	0.01901	4e-07	0.001738	3.99e-07	0.001779	2.05e-06
DS4	0.2133	5.041e-05	0.05023	5.327e-05	0.05101	0.0002019
DS5	0.4132	5.804e-09	0.07611	9.71e-09	0.07645	8.373e-05
DS6	0.7397	2.948e-07	0.1277	2.952e-07	0.128	8.608e-05
DS7	0.02859	6.743e-09	0.009177	6.615e-09	0.009194	0.001743

VI. CONCLUSION AND FUTURE WORKS

In this study, IDSs were proposed to detect network anomalies using machine learning approaches. The CICIDS 2017 [36] was used as the dataset because of its up-to-datedness, wide attack variety, and numerous network protocols (e.g. Mail services, SSH, FTP, HTTP, and HTTPS). This dataset holds more than 80 features that define the network flow. The results showed that DS and RF classifiers achieved near equal accuracies. The best performer in our study was DS and RF. The mean value of accuracy achieved by DS and RF was 0.999, the worst performance given by GNB for all datasets. We performed the paired t-test after comparing the accuracy, recall, precision, and f-score results attained by possible pairs of classifiers. By comparing the DS and RF, the p-values are larger than 0.05, accepting the null hypothesis that the mean values of both classifiers are the same. For the comparison of DS and GNB, the p-values are less than 0.05, rejecting the null hypothesis. In this analysis, a dataset comprising CSV records containing features acquired from the network flow was used as the training and test data. Unfortunately, this strategy isn't basically reasonable in a real system. However,

this problem can be solved through live network data using machine learning methods.

An interesting future work might include analyzing and studying the effects of various feature selection approaches to select the optimal set of features for building robust IDSs. One future direction is to develop an IDS based on deep learning and transfer learning approaches to deal with data sparseness issues.

REFERENCES

[1] S. KEMP, https://thenextweb.com/contributors/2019/01/30/digital-trends-2019-every-single-stat-you-need-to-know-about-the-internet/, 2019. [Online]. Available: https://thenextweb.com/contributors/2019/01/30/digital-trends-2019-every-single-stat-you-need-to-know-about-the-internet/

[2] AV-TEST, https://www.av-test.org/en/statistics/malware/, 2019. [Online]. Available: https://www.av-test.org/en/statistics/malware/

[3] A. Krepinevich, C. f. S. Assessments, and Budgetary, *Cyber Warfare: A "nuclear Option"*. Center for Strategic and Budgetary Assessments, 2012. [Online]. Available: https://books.google.com.sa/books?id=YrxSjwEACAAJ

[4] Y. Xiao, X. Shen, and D. Du, *Wireless Network Security*. Springer US, 2007. [Online]. Available: https://books.google.com.sa/books?id=efCKBrOOqXQC

[5] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *2009 Third International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2009, Conference Proceedings, pp. 268–273.

[6] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (idps)," National Institute of Standards and Technology, Report, 2012.

[7] V. Jyothsna, V. R. Prasad, and K. M. Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.

[8] H. Holm, "Signature based intrusion detection for zero-day attacks: (not) a closed chapter?" in *2014 47th Hawaii International Conference on System Sciences*, 2014, Conference Proceedings, pp. 4895–4904.

[9] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*. Australian Computer Society, Inc., 2005, pp. 333–342.

[10] H. H. Volden, "Anomaly detection using machine learning techniques," Thesis, University of Oslo, 2016.

[11] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[12] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804518303886

[13] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915023479

[14] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804515002891

[15] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer security: principles and practice*. Pearson Education Upper Saddle River (NJ, 2012.

[16] S. F. Yusufovna, "Integrating intrusion detection system and data mining," in *2008 International Symposium on Ubiquitous Multimedia Computing*, 2008, Conference Proceedings, pp. 256–259.

[17] M. Chhabra, B. Gupta, and A. Almomani, "A novel solution to handle ddos attack in manet," *Journal of Information Security*, vol. 4, no. 03, p. 165, 2013.

[18] A. Mishra, B. B. Gupta, and R. C. Joshi, "A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques," in *2011 European Intelligence and Security Informatics Conference*, 2011, Conference Proceedings, pp. 286–289.

[19] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of probing attacks," in *2009 IEEE Symposium on Industrial Electronics and Applications*, vol. 2, 2009, Conference Proceedings, pp. 557–562.

[20] E. Hodo, X. Bellekens, A. Hamilton, P. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, Conference Proceedings, pp. 1–6.

[21] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (sdn)," *arXiv preprint arXiv:1611.07400*, 2016.

[22] S. Paliwal and R. Gupta, "Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57–62, 2012.

[23] N. Ashraf, W. Ahmad, and R. Ashraf, "A comparative study of data mining algorithms for high detection rate in intrusion detection system," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 2, no. 1, 2018.

[24] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, Conference Proceedings, pp. 1–6.

[25] K. Jaekwang and L. Jee-Hyong, "A slow port scan attack detection mechanism based on fuzzy logic and a stepwise p1olicy," in *2008 IET 4th International Conference on Intelligent Environments*, 2008, Conference Proceedings, pp. 1–5.

[26] M. M. Najafabadi, T. M. Khoshgoftaar, C. Kemp, N. Seliya, and R. Zuech, "Machine learning for detecting brute force attacks at the network level," in *2014 IEEE International Conference on Bioinformatics and Bioengineering*, 2014, Conference Proceedings, pp. 379–385.

[27] M. Garcia, D. Llewellyn-Jones, F. Ortin, and M. Merabti, "Applying dynamic separation of aspects to distributed systems security: A case study," *IET Software*, vol. 6, no. 3, pp. 231–248, June 2012.

[28] K. Gupta, R. R. Singh, and M. Dixit, "Cross site scripting (xss) attack detection using intrustion detection system," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, Conference Proceedings, pp. 199–203.

[29] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey, and J. A. Halderman, "The matter of heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 475–488. [Online]. Available: http://doi.acm.org/10.1145/2663716.2663755

[30] R. Lippmann, https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, 1998. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

[31] S. Hettich and S. D. Bay, "The uci kdd archive," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[32] S. K. Sahu, S. Sarangi, and S. K. Jena, "A detail analysis on intrusion detection datasets," in *2014 IEEE International Advance Computing Conference (IACC)*. IEEE, 2014, Conference Proceedings, pp. 1348–1353.

[33] A. Özgür and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, p. e1954v1, 2016.

[34] C. I. for Cybersecurity, https://www.unb.ca/cic/datasets/nsl.html, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[35] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, pp. 357–374, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404811001672

[36] A. H. L. Iman Sharafaldin and A. A. Ghorbaniy, https://www.unb.ca/cic/datasets/ids-2017.html, 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

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

[38] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers and Security*, vol. 24, no. 4, pp. 295–307, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016740480400238X

[39] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal*, vol. 16, no. 4, pp. 507–521, 2007. [Online]. Available: https://doi.org/10.1007/s00778-006-0002-5

[40] W. Yassin, N. I. Udzir, Z. Muda, and M. N. Sulaiman, "Anomaly-based intrusion detection through k-means clustering and naives bayes classification," in *Proc. 4th Int. Conf. Comput. Informatics, ICOCI*, vol. 49, 2013, Conference Proceedings, pp. 298–303.

[41] D. P. Gaikwad and R. C. Thool, "Intrusion detection system using bagging ensemble method of machine learning," in *2015 International Conference on Computing Communication Control and Automation*, 2015, Conference Proceedings, pp. 291–295.

[42] T. H. Divyasree and K. K. Sherly, "A network intrusion detection system based on ensemble cvm using efficient feature selection approach," *Procedia Computer Science*, vol. 143, pp. 442–449, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918321136

[43] K. Dong Seong, N. Ha-Nam, and P. Jong Sou, "Genetic algorithm to improve svm based network intrusion detection system," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 2, 2005, Conference Proceedings, pp. 155–158 vol.2.

[44] K. Kumar and J. S. Batth, "Network intrusion detection with feature selection techniques using machine-learning algorithms," *International Journal of Computer Applications*, vol. 150, no. 12, 2016.