I.J. Information Engineering and Electronic Business, **2022, 2, 1-22**
Published Online April 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2022.02.01

Investigation of Machine Learning Algorithms for Network Intrusion Detection

Shadman Latif, Faria Farzana Dola, MD. Mahir Afsar, Ishrat Jahan Esha
Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: sadmanxp@gmail.com, fariadola24.aiub18@gmail.com, mahirafsar1010@gmail.com, esha.ishrat1810@gmail.com

Dip Nandi
Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: dip.nandi@aiub.edu

Received: 02 January 2022; Revised: 16 February 2022; Accepted: 05 March 2022; Published: 08 April 2022

Abstract: Network intrusion is an increasing major concern as we are rapidly advancing in technology. To detect network intrusion, Intrusion Detection Systems are required. Among the wide range of intrusion detection technologies, machine learning methods are the most appropriate. In this paper we investigated different machine learning techniques using NSL-KDD dataset, with steps of building a model. We used Decision Tree, Support Vector Machine, Random Forest, Na ïve Bayes, Neural network, adaBoost machine leaning algorithms. At step one, one-hot-encoding is applied to convert categorical to numeric features. At step two, different feature scaling techniques, including normalization and standardization, are applied on these six selected machine learning algorithms with the encoded dataset. Further in this step, for each of the six machine learning algorithms, the better scaling technique application outcome is selected for the comparison in the next step. We considered six pairs of better scaling technique with each machine learning algorithm. Among these six scaling-machine learning pairs, one pair (Na ïve Bayes) is dropped for having inferior performance. Hence, the outcome of this step is five scaling-machine learning pairs. At step three, different feature reduction techniques, including low variance filter, high correlation filter, Random Forest, Incremental PCA, are applied to the five scaling-machine learning pairs from step two. Further in this step, for each of the five scaling-machine learning pairs, the better feature reduction technique application outcome is selected for the comparison in the next step. The outcome of this step is five feature reduced scaling-machine learning pairs. At step four, different sampling techniques, including SMOTE, Borderline-SMOTE, ADASYN are applied to the five feature reduced scaling-machine learning pairs. The outcome of this step is five over sampled, feature reduced scaling-machine learning pairs. This outcome is then finally compared to find the best pairs to be used for intrusion detection system.

Index Terms: Machine Learning, Network intrusion**,** Intrusion Detection System, sampling, data preprocessing, NSL-KDD, KNN.

1. Introduction

The twenty-first century is highly dependent on internet and technologies which involves in keeping sensitive information or personal data to companies providing services. Network intrusion is a “red notice” to the service providers and consumers worldwide. Network intrusion compromises services or data of users and costs millions. To lessen this network intrusion and its effects on consumers, Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) are being developed.

Automatic and accurate detection of intrusion/attack makes an IDS easier and more reliable. Hence, using machine learning techniques can help achieve automation and more accurate detection of attack which is highly desirable in this century. The challenge here is to find the best machine learning algorithm. This challenge is worsened by the large feature set of a dataset with different datatype. The larger the number of features the greater the computation cost. Hence the achievement of any study related to machine learning algorithms application on IDS data shall be the selection of appropriate machine learning algorithm and appropriate set of features.

The objective of our research is to investigate machine learning techniques with different feature scaling, feature reduction and over sampling technique and determine the best combination of techniques for intrusion detection system. To meet the objective, we are following few steps. At step one, one-hot-encoding is applied to convert each categorical feature into new features and assign a binary value of 1 or 0 to those features. For example, an instance in NSL-KDD

with protocol_type "TCP", one-hot encoding converts the "protocol_type" feature to "protocol_type_icmp", "protocol_type_tcp" and "protocol_type_udp" and then assigns value "1" to equivalent column that represents TCP type which is "protocol_type_tcp" and assigns "0" to the rest. At step two, different feature scaling techniques, including normalization and standardization, are applied on these six selected machine learning algorithms with the encoded dataset. Further in this step, for each of the six machine learning algorithms, the better scaling technique application outcome is selected for the comparison in the next step. For example, normalization increased the accuracy, precision and recall with an increase in train time and prediction time for KNN. Using standardization did not have significant improvement on the performance. So, for KNN we selected the pair with normalization. We considered six pairs of better scaling technique with each machine learning algorithm. Among these six scaling-machine learning pairs, one pair (Naïve Bayes) is dropped for having inferior performance. Hence, the outcome of this step is five scaling-machine learning pairs. At step three, different feature reduction techniques, including low variance filter, high correlation filter, Random Forest, Incremental PCA, are applied to the five scaling-machine learning pairs from step two. Further in this step, for each of the five scaling-machine learning pairs, the better feature reduction technique application outcome is selected for the comparison in the next step. For example, correlation filter increased the accuracy and recall slightly for the pair with KNN compared to other feature reduction techniques. So, for KNN we selected the pair with correlation filter. The outcome of this step is five feature reduced scaling-machine learning pairs. At step four, different sampling techniques, including SMOTE, Borderline-SMOTE, ADASYN are applied to the five feature reduced scaling-machine learning pairs. The outcome of this step is five over sampled, feature reduced scaling-machine learning pairs. For example, using Borderline SMOTE as oversampling technique for the pair with KNN, the performance of the model increases with no overfitting problem. So, for KNN we selected the pair with Borderline SMOTE as oversampling technique.

This paper is organized in seven sections following the Introduction. Section 2 provides details about network intrusion. Section 3 discusses about network intrusion detection. Section 4 discusses the related works of machine learning and IDS. Section 5 discusses related works of machine learning and hybrid IDS and a brief description about different machine learning techniques. In Section 6 we present a comparative analysis on different machine learning techniques. In Section 7 we briefly concluded the analysis and stated our future work.

2. Network Intrusion

Intrusion is the unauthorized access to something that is sensitive to someone which implies a threat to their personal belongings. Network Intrusion [1] is seen as a breach in privacy and is considered a major concern over the past decade. In terms of cyber security, intrusion is referring to as network intrusion and is define as the unauthorized access or activity in a network domain. It involves in breaking into computers through assigned domain and stealing valuable information and resources and jeopardizing the network security. Network Intrusion can be passive and active. Passive is gaining access illegally and stealthily and without getting detected. Active intrusion affects the network resources and bandwidth. Such unauthorized access can compromise the CIA triad [2]. Network intrusion can be of many types including malwares, DoS, MITM and more discussed in the next section.

2.1. Types of Intrusion and effects

Because of increasing activity taking place in the growing interconnected digital world, it is becoming difficult to point out anomalies that can indicate in an occurrence of network intrusion. Network intrusion techniques can be of many types [1,3], it can be from outside or inside the network structure.

Multi Routing attack, where the attacker uses multiple routers as a leverage to illegally gain access to victim's network. Its only feasible when the networks use asymmetric routing and is configured for multi-routing.

Living off the land attack is when an attacker uses existing tools or processes or even stolen credentials to compromise a network and steal valuable information over the time.

CGI Scripts attacks, although less popular today, is an easy way to compromise a network if it does not require input verification or scan for backtracking. As CGI uses servers to pass user request to and receive data back from application allowing an easy opening for attackers to access network system files that can be used to access files that should not have been accessible via web.

Protocol-specific attack, also known as Man-In-The-Middle (MITM) attack [4], is a cryptographic attack that illegally impersonates different protocols and gain access to data which would not have been normally accessible. The attacker invisibly intrudes in the communication between two nodes, conventionally a client and an application, spoofing and stealing valuable information including login certifications, account details, etc.

Traffic flooding attack, attacks by generating huge traffic load, making the traffic too large for the system to screen adequately, which allows the attacker to execute an attack without getting detected. Distributed DoS attack [5] is a traffic flooding attack that compromises the victim's resources, then triggers the attack. Recently, cloud computing has been the most affected victim of the world wide web.

Malwares [6] including Trojans, worms, viruses, etcetera can be the reason for network intrusion. They are malicious software that compromises the victim. While each type of malware is different than the other, the damage

they incur is the same, and this damage is hijacking the victim's resources and private information. Most of the malware is found in the windows platform that leads to the most breach in private information than any other platform. But with the increasing demand in web services and cloud computing, the growth rate of malware in this platform is growing much significantly than any other platform. This leads to a conclusion that web services are becoming the most vulnerable place for a victim and his/her personal information being attacked and compromised.

In 1988, about $100 million damage was done by Morris worm to 60 thousand computers connected to internet. Within five years of that attack, about 400 thousand computers got infected. In 2011 Anti-Spyware attacked different versions of windows [1]. In 2013, Target an American retail corporation, lost 70 million customers personal information to a cyberattack [7]. It resulted in a market loss of $890 million in stock and $292 million for the expenses of data-breach. Almost $3/4^{th}$ of the attacks analyzed in [7] resulted in financial information loss. In 2017 the number of attacks increased 7 times from it was in 2005 and most of the attacks were on service industries followed by finance, manufacturing and retail industries. Not only it can harm firms and companies with resources, but it can also attack individuals. Information breach cause sexual assault, financial fraud [8] which can lead to depression, anxiety, and disorders etc. Findings in [8] sates that 9 out of 100 people finds it 'very' safe online. 2017 ransom attack 'WannaCry' affected many sectors including public, healthcare organizations, telecommunication companies, education sectors, etc. This resulted in organizations closing, people losing jobs, production stopping, and it resulted in $8 billion economic loss globally. In 2015, many banks in UK faced a DoS attack and in 2017 they faced a same type of attacks [8]. In 2013, a massive DDoS attacks clocking at 300Gbps targeted Spamhaus, an industry-leading organization for spam filtering. This attack was the biggest DDoS attacks of that time and cause network disruption through all of Britain affecting other companies in crossfire. In 2018, GitHub was attacked with 1.3 Tbps DDoS attack with 126.9 million packets of data each second. The attack lasted for about 20 minutes because GitHub uses a DDoS mitigation service. Later in 2018, right after the GitHub attack, one of the customers of NETSCOUT was attacked with a 1.7 Tbps largest of that time. In 2020, surpassing the 1.7Tbps attack, Amazon Web Services (AWS) was attacked with 2.3Tbps resulting in three days of service blockage of some of AWS's customers.

All types of network intrusion have a major effect in the field of computer science as it did a decade ago and progressively increasing its severity in sabotaging and damaging valuable information and resources to individuals as well as companies. Every year, it not only costs privacy but also millions of dollars. Large companies, even though adapted to latest cybersecurity technology, are getting attacked by hackers which is costing them their customers as well as market reputation. As most of the local services are turning to cloud services, cloud computing and services are becoming increasingly popular as well as their concerns for privacy. Most of the hijacking and attacking are happening over the cloud and shockingly most of the attacks are getting away undetected. We should focus on finding and developing new ideas to detect and prevent these attacks. So, intrusion detection is as important as other advanced industrial technologies and new methods of intrusion detection should be developed with upmost importance to try to prevent the inevitable cyber-attacks now and in the future.

3. Intrusion Detection

Intrusion detection [9] is to identify different intrusion in network traffic flow in real-time. The main goal is to detect any external or internal unauthorized intrusion in systems. Because of the increment of network users as days passes by, network intrusion detection has become challenging. Attackers and intruders are hijacking valuable information and resources while avoiding identification. Intrusion detection Systems are being developed using variety of techniques and recently, over the past decade researchers have advanced in network intrusion detection.

CIA triad for cybersecurity is data confidentiality, integrity, and availability. The core purpose of Intrusion Detection System (IDS) is to detect attacks that implies a threat to the triad. IDSs are security tools to protect users from unauthorized disclosure when using the network or over communication. Sometimes, network intrusions are not completely preventable, and it is preferred to detect attacks as accurately as possible to comply with the network services. So, IDSs are used to notify victims by alerting them about the attack in real-time, so that preventive measures can be taken. Intrusion Prevention systems (IPS) are extended version of IDSs with prevention techniques. To develop an IPS we first need an IDS. The intrusion prevention is not possible without its detection. So, it is more important to detect attack accurately first and then provide the prevention. As IPSs are not the scope of the study, we will be focusing on IDSs.

3.1. Intrusion Detection Techniques

Intrusion detection systems (IDS) [10,11,12] are security software and sometimes with hardware, used for detection of malicious activity that can cause system security issues. Intrusion detection systems uses techniques [3] that are generalized into two types, namely Signature-based intrusion detection (SID) and Anomaly-based intrusion detection (AID) [11,13,14,15,16,17].

1) *Signature-based intrusion detection (SID):* Signature-based or knowledge-based intrusion detection is also known as Misuse Detection. The concept of Signature-based intrusion detection system (SIDS) is to compare the pattern of an attack to predefined knowledge and known signatures. Signatures including source address, destination

address, ports, etc. Because of signatures being known, these types of detection systems usually have high accuracy against previously known attacks. But as it creates signatures from new attack variations every time, the computational performance degrades as new signatures keeps on increasing. The larger the signature database, the more accurate the system, but less efficient. The main advantage of using this type of system is that the database can be updated with new attack types of latest detections. So, it can detect attacks efficiently without significant false alarm rate. However, this type of system has a major drawback. For a zero-day attack or a variation of a known attack, the system will not be able to detect the attack. The reason behind this inability is SIDSs are only based on regular expression and string matching. Therefore, it is only effective in detecting known attacks.

In cloud computing, SIDSs can be used both in front end and back end for detecting attacks externally and internally respectively. As increase in demand of cloud computing and increase in zero-day attacks, SIDSs have become less effective because of no prior knowledge on these zero-day attacks. Also, polymorphic variants of malwares can undermine the adequacy of SIDSs even more. The remedy for this main disadvantage is Anomaly-based intrusion detection system (AIDS) which is discussed in the next sub-section.

2) *Anomaly-based intrusion detection (AID):* Anomaly-based intrusion detection is typically modeled using machine learning as well as statistical-based methods and knowledge-based methods. The model is usually a normal system behavior in network and if the observed behavior has any significant deviation from the original model, it is considered as an intrusion. AIDS compares features including count of emails sent, count of failed user logins, etc. Malicious activity has different behavior than normal activity, so this assumption is used to develop this type of systems. Anomaly-based intrusion detection systems (AIDS) are developed in two phases, training, and testing. The main advantage of AIDSs over SIDSs is that AIDSs can detect zero-day/novel attacks as well as new variety of know attacks. But this advantage comes with a cost of having greater false alarm rate. With recent research, it can also be possible to identify the attacker with AIDS.

In cloud computing, AIDS can detect attacks and unknown anomalies at different level including network level and system level. Because huge traffic flow occurs in different level, it is difficult for AIDS to monitor and detect intrusion efficiently. AIDS can have high false positive rate because certain anomalies detected can be new normal behavior instead of being an attack or intrusion.

Both Signature-based and Anomaly-based intrusion detection technique can be used to develop intrusion detection system. Each technique has its own advantages and disadvantages. The findings by reviewing intrusion detection techniques are: AID technique can be beneficial when used in IDS for networks and SID technique can be beneficial when used in IDS for computers and hosts. The next subsection discusses different types of intrusion detection systems using SID and AID techniques.

3.2. Intrusion Detection Systems

There are three common types of IDSs namely Network Intrusion Detection System (NIDS), Host Intrusion Detection system (HIDS) and Hybrid Intrusion detection system (Hybrid IDS).

1) *Network-Based Intrusion detection system (NIDS):* Network-Based Intrusion detection system (NIDS) detects cyber-attacks, computer denial (DOS) attacks, malware, or port scans on computer networks. NIDS identifies malicious activity by monitoring network traffic and recognizing suspect patterns on incoming packets. Every malicious movement is usually reported to the Administrator or Security Information and Event Management (SIEM) system. NIDS can include both AID and SID intrusion detection techniques. Nowadays, a lot of traffic is encrypted, and it is invulnerable for analysis by a network-based IDS. The network intrusion detection system is unable to monitor encrypted traffic. Only the part of the network to which an IDS is attached can detect attacks. If suspicious traffic travels to a different part of the network, NIDS cannot detect it. NIDS monitors the network's total traffic from which the hosts are connected, receives data, and makes decisions. NIDS is affordable and provides instant real-time detection of network attacks to minimize the possibility of network damage due to intrusion activities. Since network-based detection focuses only on network traffic, such systems do not know about the malicious activity that can occur on any host in the network. A signature-based NIDS watches network traffic to detect unusual patterns in data packets and signs of familiar network intrusions. Signature-based NIDS uses a database of available intrusion types and data patterns to quickly detect intrusions and allow appropriate action to be taken once identified. Anomaly-based NIDS uses the system's baseline to track unusual or suspicious activity on the network normally. This method takes time to set up because NIDS needs to learn about your usage patterns [18].

2) *Host-Based Intrusion detection system (HIDS):* Host-Based Intrusion Detection System (HIDS) monitors or analyzes log files and detects malicious or intrusive activity on a machine [19]. HIDS makes their decisions based on information they receive from a single user. HIDS inspects specific actions such as access of files, usage of applications and information. It tracks any changes made in regular activities. In Recent years, several forms of HIDS have been created that can be used to monitor networks. HIDS determines whether a system has been compromised and alerts the administrators concerned [20]. Host-based intrusion detection systems runs on individual system, including data collection and analysis techniques on a specific system [21,13]. A HIDS can be further classified according to the technique used for intrusion detection, namely, signature-based detection and anomaly-based detection techniques discussed in previous subsection. Both uses data extracted from the target of the analysis to determine whether

infiltration has occurred [22]. Though HIDS and antivirus monitors all activities inside a system, HIDS works differently from anti-virus. Anti-virus is not enough to identify and analyze system-specific attacks including memory leakage, buffer overflow attacks on memory and other operating system errors. But HIDS collects and analyzes system data such as file system status, system call patterns, and any inconsistency in identifying system events [23]. HIDS system uses audit trail information and system logs to identify malicious actions inside the system. HIDS can monitor access to user-specific information, which is a significant advantage [21,24]. HIDS can identify inappropriate use of company assets. If the activity pattern is similar to a past attack, the activity can be stopped with that company's resources, thus preventing the attack. HIDSs can deal with internal threats and abnormal behavior on the local network as well. The HIDS can monitor and respond to specific user actions and file access [25]. Some notable disadvantages are: i) They cannot see network traffic [21]. ii) HIDS relies heavily on audit trails that can drain a lot of resources and space on the server and iii) lacks cross-platform inter-operability. Although many improvements have been made to the intrusion detection method, for many applications the false-positive rate is not low enough. HIDS requires more manual input to reduce false-positive alarms [25]. If HIDS is designed efficiently, it can withstand attacks such as outgoing DoS attacks and notify the system administrator that the system or its resources are endangered [26]. Data trace and basic system features are used to detect any malicious activity on the host system. This method helps to recognize doubtful and suspicious actions but does nothing to prevent them. For prevention we need an Intrusion Prevention system.

3) *Hybrid Intrusion detection system (Hybrid IDS):* Hybrid IDS is a mixture of two or more different detection techniques. Conventional Hybrid IDS are SID and AID stacked in series or parallel detection. It is more effective than other stand-alone detection systems like NIDSs and HIDSs.

NIDSs and HIDSs use one of the detection techniques from SID and AID. Therefore, NIDS and HIDS inherits the drawbacks of SID and AID. For systems using SID technique they have three drawbacks. Firstly, it is easy for attackers to trick SIDS with polymorphic behavior of a malware subsequently giving them chance to break into computer systems. Secondly, the larger the signature database gets, the greater time it takes for the system to analyze and detect, making it inefficient. Lastly and importantly, SIDS cannot detect zero-day attack for there is no signature match with the database of new attack type. AIDS overcomes the problem of SIDS having strong generalization capability and to detect new attacks. But AIDS gives large amount of false alarm due to changing cybersecurity scenario. Both SIDS and AIDS can model using statistical-based method, knowledge-based method, and Machine learning [15,16].

Hybrid IDS uses both SID and AID technique to detect attacks better. Using Hybrid IDS, we can minimize the drawbacks from stand-alone SIDS and AIDS. Hybrid IDS can detect new attack types efficiently with minimized false alarm, by stacking both SIDS and AIDS for detection [14,27,28,29,30,31,32]. The network packets first are sent to SIDS for anomaly detection. If there is no intrusion detection, the packets are forwarded to AIDS for anomaly detection. If the packets are malicious, it alerts the user and labels the data as anomaly and stores in both SIDS and AIDS database. Because of rising threat in cybersecurity, new attacks are happening every day, it is necessary to detect as much attack as possible efficiently. Hybrid IDS though uses more computational power than usual standalone SIDS and HIDS, it is necessary for efficient and accurate intrusion detection. As computational power of devices is increasing at every innovation, it should not be a factor for new devices to support powerful Hybrid IDS. But it increases the cost for having a secured device with powerful Hybrid IDS. It is required to develop powerful Hybrid IDS which consumes less computational power and give better and accurate detection efficiently. This can be achieved by doing a comparative analysis on different ML techniques to develop a model/classifier.

Summarizing this subsection, Network Intrusion Detection, Host-based Intrusion Detection, Hybrid Intrusion Detection systems are the common types of network intrusion detection system. Each IDS has its own advantages and drawbacks compared to other IDSs. But Hybrid IDS is type of IDS that harnesses the advantages of NIDS and HIDS while bypassing the drawbacks for optimum performance. These IDSs are modeled using different attack datasets with variety of known attacks accumulated and developed by different organizations. The following subsection discusses different IDS dataset used to build IDSs.

3.3. Datasets used for Intrusion Detection system

Intrusions are sets of actions that compromises confidentiality, integrity, and availability of data. Cyber-attacks/Network intrusion is categorized into four types [33] including Denial of service (DoS), Remote to Local (R2L), User to Root (U2R) and Probe.

1) *Denial of Service (DoS):* DoS is a type of attack that attempts to block access to targeted sites for legitimate users [34]. It makes network resources too busy or full to manage these user requests by flooding the servers with illegitimate requests. Usually, the attacks are targeted upon network services including emails, ecommerce, web application, etc. This results in temporary unavailability of connectivity and services. Some common DoS attacks are TCP SYN flooding, teardrop, ping of death, buffer overflow, smurf, etc. A distributed DoS (DDoS) is same as DoS but DDoS uses multiple machines or computers to attack the target by depleting bandwidth or resources. There are different DDoS attack including Volumetric Attacks, TCP Handshake/SYN Floods, Multi-vector DDoS Attacks, Application Layer Attacks, etc. Multi-vector DDoS attacks is the most effective and difficult to counter among the DDoS types. Preventing DDoS attacks are difficult because it’s hard to differentiate between attacks and legitimate requests.

2) *Remote to Local (R2L):* R2L is a type of attack where packets are sent by attacker to targeted remote machine despite having no account on the machine [34]. It enables the attackers to gain access to the machines as a local user or root user and conduct malicious activities. Most common R2L attack is buffer overflow.

3) *User to Root (U2R):* U2R is a type of attack where attackers gain illegal access as an unprivileged local user of the targeted machine [34]. This unauthorized and illegal access may be conducted through exploits including social engineering, sniffing password, dictionary attack, etc. These exploits allow the users to gain root access to victim's machine.

4) *Probe:* Probe is a type of attack that scans the network to identify valid addresses and to collect information of individual users/host. For example, OS information and other services running in the user's machine. Few common probing attacks including Nmap, portsweep, IPsweep, etc.

To model an IDS, we need proper dataset with all types of attacks that can sufficiently train a classifier for proper attack detection. Some of the most widely used dataset for developing IDSs are KDDCup99, NSL-KDD, UNSW-NB15 and ADFA. Machine Learning algorithms discussed later in Section 4, have been widely used to analyze and classify data in these datasets for IDS.

KDDCup99 also known as KDD99 is the original IoT network intrusion dataset from 1999. KDD99 contains attacks including DoS, R2L, U2R and Probe [35]. The dataset is split into two subsets including training and testing. The attack class distribution of KDD99: a) Normal- 97278(training), 60593(testing). b) DoS- 391458(training), 222200(testing). c) R2L- 1126(training), 5993(testing). d) U2R- 52(training), 39(testing). e) Probe- 4107(training), 2377(testing). The count of attacks type U2R, R2L, Probe is less than 1% in respect to the whole dataset. The main drawback of this dataset is the imbalance data class distribution because of the small number of R2L, U2R and Probe attack [36]. The dataset comprises of 41 feature attributes and class label for all the attack types.

NSL-KDD is the improved version of KDD99 dataset released in 2009. The redundant instances from the KDD99 were removed to create NSL-KDD a more efficient intrusion dataset [36]. Like KDD99, NSL-KDD is also split into two subset including training and testing with 41 feature attributes and class label of all the attack types. There are two training datasets "KDDTrain+" and "KDDTrain+20%". The attack class distribution of NSL-KDD considering KDDTrain+: a) Normal- 67343(training), 9711(testing). b) DoS- 45927(training), 7458(testing). c) R2L- 995(training), 2754(testing). d) U2R- 52(training), 200(testing). e) Probe- 11656(training), 2421(testing). Although there is still class imbalance, but the instances of normal and DoS classes are less compared to KDD99 dataset which somewhat solves the imbalance problem. But to properly fix this problem, so that the error rate of IDSs is lower, down sampling the majority class or oversampling the minority class can be done [30].

UNSW-NB15 dataset is a hybrid of modern normal and contemporary attacks released in 2015 [35]. It contains 42 feature attributes and class label. The attack classes in UNSW-NB15 are Normal, Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, and Worm. The training set has 175,341 instances and the testing set has 82,332 instances.

ADFA dataset is a cyber security benchmark dataset that can be used to build IDS [27]. It has two versions including ADFA-LD for Linux and ADFA-WD for Windows operating system [11]. For ADFA-LD, the total traces are 5951 out of which 746 attack data, 4372 testing data, 833 training data. For ADFA-WD, the total traces are 7724 out of which 5542 attack data, 1827 testing data, 355 training data.

In the above subsections, Intrusion Detection (ID) techniques and Intrusion Detection systems (IDS), we have discussed different techniques including Signature Intrusion detection (SID) and anomaly intrusion detection (AID) techniques that can be used to build several types of intrusion detection systems including NIDS, HIDS and Hybrid IDS. One of the major disadvantages of SID based IDS is that they are vulnerable when comes to detecting zero-day attacks. And as the database of attack signatures gets bigger, the more time it takes to detect new attacks. This problem can be mitigated when using AID as detection technique, but the major disadvantage of AID is its high computational power requirement and more false alarm rate, which is not ideal for an IDS. NIDSs, HIDSs and hybrid IDSs based on these SID and AID, inherits all their drawbacks and limitations. So, to try to mitigate the limitations we are going to use the concepts of machine learning to build models for IDS using NSL-KDD dataset discussed in this section.

4. Machine Learning

Machine learning (ML) uses artificial intelligence to make a system qualified for learning and progressing in an automatic manner from expertise except of being programmed in an explicit manner. Machine learning means a method or way for machines to learn intended things without explicitly programming it [37]. There are four main type of machine learning techniques including supervised, unsupervised, semi-supervised and reinforcement learning. These machine learning techniques are used to develop a model in two phases, training, and testing. The training phase is done by training the model using specific dataset, with labeled or unlabeled data. The testing phase is done to evaluate the model, where the testing data is part of the dataset used for training. A more complex and profound version of machine learning is deep learning, which is modeled after human brain and is very useful in lots of sectors. In the growing flow of applications for machine and deep learning techniques, we are applying a variety of combinations of algorithms in

diverse number of areas including cybersecurity, facial recognition, disease detection [37], etc. But machine learning can be very costly. Machine learning process is about identifying suitable datasets and prepare them for analyzing. Then the appropriate machine learning algorithm is needed to be chosen. Based on the selected algorithms an analytical design model should be made so that this model can be tested on various data sets.

Two most common machine learning techniques are supervised and unsupervised learning. Supervised learning is generating functions/model to map a set of input to a known output. Then this model can be used for predicting the output for new set of inputs. Unlike supervised learning, unsupervised learning does not have a known output to generate a function/model. For supervised machine learning algorithms, we have K-Nearest Neighbor (KNN), Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB) [38,39], etc. For unsupervised machine learning algorithms, we have clustering, K-Means, Principal Component analysis (PCA), Boosting, Bagging, Stacking and for deep learning we have several types of Neural Networks [38].

As machine learning method is an automated way of predicting certain things, but beforehand we need to train specific models for specific action. A model of certain dataset can be trained with different machine learning algorithms, each model can have different outcome than other. So, a comparative analysis can provide an improved perspective on the best working techniques/algorithms in a certain application area. The remaining of this section explains the idea of a comparative analysis of intrusion detection systems with different ML algorithms over the last decade.

In paper [40] the authors used Bayesian Network (BN), Decision Tree (C4.5), Naïve Bayes (NB) and Naïve Bayes Tree (NBT) to develop models for network traffic flow classification. They compared the models using computational performance along with other outcomes. They found that DT (C4.5) has the fastest classification speed as well as good accuracy which will be best for real time traffic flow classification.

In paper [41] the authors compared more ML algorithms for network intrusion detection including Bayesian Network, Logistic, J48, JRip, IBK, PART, Random Tree, Random Forest and REPTree. They also used Boosting (adaBoost), Bagging, Bending (stacking) to boost the performance of the classifiers. They used NSL-KDD dataset for developing the system. The study concluded that Random Forest and Bayesian Network with the highest accuracies and low training time, is suitable for network intrusion detection.

In paper [42] authors compared Decision Tree (C4.5), SVM and a deep learning method Neural Network for Intrusion Detection System (IDS). They used KDD-99 dataset for developing the system. For attacks including Probe, U2R, DoS attacks, DT (C4.5) outperformed the rest. The result of the study pointed that DT has the highest accuracy and detection rate for these attacks and is best for an IDS.

In paper [43] the authors used Naïve Bayes, SVM, SVM-RBMs, Decision Tree (C4.5) to develop models and compare their accuracy for network intrusion detection. SVM-RBMs is similar to deep learning, with multiple hidden layers in between input layer and output layer. They used KDD-99 dataset and used SMOTE, an over-sampling technique, to solve the imbalance problem of the number of attacks in each attack type. They concluded that with SMOTE applied, SVM-RBMS has the best precision. They also stated deep learning has advantages for big data and analysis because of its capability to adapt to changes.

In paper [44] the authors compared J48, Random Forest, Random Tree, Decision table, MLP, Naïve Bayes, Bayes Network for developing an intrusion detection system. They used KDD-99 dataset for testing and training. They concluded the study that Random Forest has the highest accuracy among all the ML algorithms used. They stated that for network availability and confidentiality, the number of falsely predicted instances should be considered.

In paper [45] the authors compared Naïve Bayes, Bayes Net, Logistic, Random Tree, Random Forest, J48, Bagging, OneR, PART, ZeroR for intrusion detection using NSL-KDD dataset. The result concluded as Random Forest, Bagging, PART and J48 being the top 4 classifier with Random Forest being the best. But because of time consumption in building the models, the authors compared feature selection and reduction method to reduce time consumption. The authors found that discretize filter in conjunction with filter method was best for reducing time. They stated that reduced time is important when considering for attack detection. However Random Forest achieved the same accuracy with lower build time, without using the feature reduction techniques.

Ensemble learning play a huge role in increasing the accuracy of IDSs for certain classifiers. In paper [46], the authors compared boosting, bagging, stacking with Naïve Bayes, iBK, J48, Jrip for SIDS. It resulted in increasing the accuracy of Naïve Bayes a lot when using boosting, while for iBk, J48, Jrip the result accuracy remained around the same. It also decreased the false positive in the same way. Bagging did not have much influence over increase in accuracy and decrease in false positive rate. Stacking gave a high accuracy result as well as low false positive with all the combination. When testing with new data, none had good accuracy and low false positive rate. Only iBK stacked with other algorithms gave the highest output of 67.9%. Boosting, bagging, and stacking also increase the execution time of certain classifiers.

In this section, few comparisons of ML algorithms conducted by different authors are discussed. Different ML algorithms are used to model different proposed intrusion detection system. The authors compared this combination of algorithms to study the performance difference in each technique. Most common algorithms compared are SVM, Decision Tree, Random Forest, Naïve Bayes, Bayesian Network. Most of the concluded result have a high accuracy. It is apparent that Intrusion detection system works well with ML algorithms. But as different algorithm gives different performance outcome, more and latest ML algorithms needs to be compared. As deep learning techniques are advancing

in Datamining, AI, Data science etcetera, it should be compared with other powerful ML algorithms. Hybrid IDS is better than NID and HID. Hence, more focus should be considered in developing Hybrid IDS with latest and powerful ML algorithm that shall have low false detection rate and faster training and classification time. This can be achieved by comparing these algorithms.

5. Machine Learning and Intrusion Detection System

Machine Learning techniques are widely used in IDS [34]. Different widely acknowledged IDSs [10] including ComputerWatch, Discovery, HAYSTACK, IDES, ISOA, MIDAS, Wisdom and Sense have used Machine Learning and Data mining techniques for attack detection. Current IDS lacks speed, have low rate of real time detection accuracy, about IPV6 they lack support. So, different ML algorithms have different impact on IDSs along with their attack detection. The more powerful and accurate the ML algorithms are the better they are at detecting different attacks and malwares. Existing IDS methods sustain from false alarms due to reason of flagging legitimate user as intruder and lot of intrusion remaining undetected. When modeling an IDS, we need to consider the amount of false positive and negative attack detection rate along with high accuracy and detection rate and faster detection time. The less the amount of false positive on normal data and false negative on attack data, the better the IDS can be considered. But an IDS probably cannot have any false detection at all. So, we must focus on developing an IDS with low false detection, high accuracy, and faster detection time. As we have discussed the advantages of Hybrid IDS over NIDS and HIDS that are using signature-based techniques or anomaly-based technique, it is justifiable to focus more on improving Hybrid IDSs. The following subsection discusses some of the research regarding hybrid IDS.

5.1. Hybrid Intrusion Detection System related works

In paper [29], the authors proposed a Hybrid IDS combining Packet header anomaly detector (PHAD) and Network traffic anomaly detector (NETAD) with SIDS Snort. PHAD and NETAD are anomaly-based approaches added to the preprocessor along with SNORT. PHAD is a network-based anomaly detection system but different in two ways. Firstly, it models protocols instead of user behavior. Secondly, it uses a time-based model for quick change in network statistics in a short time. PHAD analyses packet headers for anomaly detection. PHAD reduces false alarm rate by flagging the first anomaly. NETAD also models packets like PHAD. NETAD works in two phases. Firstly, distinguishing session beginnings by filtering incoming client sessions keeping the anomaly traffics for the next phase. Secondly, the modeling phase. The filtering of 99% of traffic from first phase makes the modeling phase simpler. SNORT is a signature-based NIDS. SNORT allows change in rule to the IDS. Every rule has a rule header and rule options. Rule header includes rule action, end-to end source and destination information, direction of traffic and protocol type. Rule option includes various conditions to help deciding on if the traffic has rule action. The authors used IDEVAL dataset for performance evaluation. When using just SNORT, it detected 27 attacks from the 201 attacks in IDEVAL dataset. When using SNORT+PHAD the number of attack detections increased to 51. When using the proposed Hybrid IDS (SNORT+PHAD+NETAD) the number increased to 146. They stated that it is more powerful than standalone SIDS and AIDS and improves on the disadvantages SIDS and AIDS have. The authors in paper [32] used the same concept of using SNORT as SID but for AID they implemented back propagation network (BPN). BPN has good attack detection capability but slow detection speed. The proposed framework can be used for both front-end and back-end of a cloud, but the modules should be positioned accordingly. In the SIDS the captured packets are matched with the SID database using SNORT for intrusion detection. If there is no intrusion detected, the packets are forwarded to AIDS. The AIDS then predicts the class based on the model built during training phase. If any attack detected, it denies the packets to be accessed and alerts the user for intrusion. The malicious packet is sent to central log database, so that it can be easily detected by SNORT at other hosts and reduces computational cost and detection time over the cloud. Though the authors kept the implementation of this framework for future research, they claimed the design to have high detection rate and accuracy as well as low false positive and negative rate and affordable computational cost.

In paper [30], the authors proposed a similar approach like paper [29], but instead they used random forest algorithm as a classifier for SID and k-means algorithm as a classifier for AID. The model proposes solutions to few problems. a) Network connection has important categorical features, but k-means only works with continuous features. b) K-means bias towards highest scale features because of continuous features on different scales. c) Some real activities are like intrusions, which can lead to high false positive rate because of normal data being in the same cluster as intrusion. The proposed Hybrid IDS framework consists of 2 phases, offline and online. Offline being the training phase and online being the testing/classification phase. They used the KDDCup99 dataset. The online phase is the part of SID, which forwards the attack features to random attack selector part of AID, if any intrusion is detected and alerts the user. If there is no intrusion detected in SID, the data is sent to AID for intrusion detection. Some components of SID and all components of AID is part of offline phase. The offline phase of SID is to build a classifier for the online phase of SID for intrusion detection. The offline phase of AID is used to classify and store the intrusions in AID database. The k-means algorithm is used to cluster the normal and intrusion data into individual clusters. If the AID detects any intrusion the connection data is sent to SID database to store the information for further SID detection. The

proposed SID only approach has detection rate of 92.73% and a very low false positive rate of 0.54%. The AID only approach has the highest detection rate of 99% but quite bad false positive rate of 12.6%. The proposed hybrid IDS framework strengthening the advantages of SID and AID achieving a detection rate of 98.3% and false positive rate of 1.5%.

In paper [47], the authors proposed a Hybrid IDS that takes advantage of the preprocessing of Support Vector Machine (SVM) and the optimization of Genetic Algorithm (GA). SVM is a ML algorithm used to preprocess data from KDDCup99 dataset. GA is then used to optimize the dataset. After testing and training they compared the accuracy, False/True Alarms, mean squared error of the hybrid IDS with SVM based IDS. The accuracy of the hybrid IDS is 98.33% over 94.8% of SVM only. The false positive and false negative rate is lower for the hybrid IDS. They stated that the performance of IDSs improves when using multiple algorithms rather than single primary ones. Like the author in paper [47], the authors in [48] used GA to optimize the data sampled by iForest. Then used RF model for prediction like the author in paper [30]. The feature selection method of GA is used to find the optimal feature set both for testing and training data. Though the proposed model did not use SID and AID, but for detection it had higher accuracy than any standalone ML algorithm used.

With similar concept of combining SID and AID in paper [29,30], the authors in paper [31] used Decision tree (C4.5) for the SID and 1-class SVM for AID. 1-class SVM can be very sensitive to training data and can increase false positive rate. So, the training data is decomposed into subsets of training data to develop multiple models, but it can be less flexible. The decomposition of training data is done according to the tree model generated from DT (Data in the same leaf belong to the same subset and for each normal leaf of the decision tree, an anomaly detection model is build using the 1-class SVM). Because conventional SVM model tends to narrow down normal data, it can cause high false positive rate. So, a high value of parameter γ can reduce the narrowing down of data which increases the detection rate and decrease the false positive rate. The proposed model (DT-SVM) focuses on smaller subsets of normal data and any outlier results in detection of attack. The result of the experiment shows that for γ=1 it has a high detection rate and low false positive rate for both known and unknown attack when compared to conventional Hybrid IDSs and only AIDSs. It also has lower testing and training time than conventional Hybrid IDSs.

In paper [27] the authors follow the same approach as [31] combining a DT algorithm for SIDS and 1-class SVM for AIDS. The Hybrid IDS approach has an online phase and an offline phase. In the offline phase, the C5 algorithm is used to update the signature database and detect intrusion/attack. The online phase is part of AIDS where 1-class SVM classifier is used to create a model for real-time intrusion detection. They used ADFA dataset that have two version, one for Linux (ADFA-LD) and one for windows (ADFA-WD). They also used NSL-KDD dataset. The result shows for SIDS only the accuracy is good but false positive rate is worse for NSL-KDD dataset. AIDS accuracy is worse as well as false positive than SIDS or the Hybrid IDS. The hybrid IDS has the best accuracy and a very low False positive rate.

In paper [49] the authors proposed an artificial neural network model for network intrusion detection. The proposed ANN model consists of a Feed forwarding network and pattern recognition. For training the feed forward Neural Network, Back Propagation Algorithm (Bayesian Regularization) was used. For training the pattern recognition Neural Network, Back Propagation Algorithm (Scaled Conjugate Gradient) was used. They used KDD CUP99 for testing and training. They used Accuracy, Mathew's Correlation Coefficient, R-squared and Mean Squared Error as performance metrics. For both of the proposed models, the performance was quite similar. The author stated that both the model could be improved and further tuned and used for more diverse intrusion dataset.

In paper [14] the authors combined Neural Network (NN) for SID and Fuzzy logic for AID. The neural network model will generate rule sets that predict intrusion and is fed in Suricata IDS/IPS for preventing attacks. The fuzzy logic model is used to detect anomaly attack, which is not on the rule set database. The proposed framework resulted in an average accuracy of 96.11%.

5.2. Widely used Machine Learning Algorithms

Machine learning techniques are widely used in developing models for intrusion detection. These techniques have two phase including training and testing. Training phase build the classifiers/model for the IDS. Testing phase determines the strength of algorithm. Some of the Hybrid IDS framework proposed by different authors discussed in the previous subsection. Some of the common machine learning techniques used are Support Vector Machine, Decision Tree, Random Forest, K-Means, Genetic Algorithm, Back Propagation Network, Fuzzy Logic, Boosting, Bagging.

1) *Decision Tree (DT):* Decision Tree is a ML technique where every possible outcome of decision is illustrated using a tree structure (top-down). The method is called branching method [34]. The tree is formed using three elements including decision nodes, branches, and leaf nodes. Decision node represents the if-then condition to classify an object. Each branch represents a possible value of an attribute. The leaf node represents the class of the object. Some common DT algorithms are C5.0, C4.5, ID3, CART, etc. DT is used to solve both classification and regression problems. DT has some advantages in problems where a) Instances can be attribute-value pairs where every possible value of each attribute can be disjoint. b) Targeted functions have discrete option such as yes or no. c) there is error in training data. d) There are missing attribute values in training data. DT performs better than any other standalone classifiers because of it just uses entropy and information gain for feature selection. The main issue is locating the attribute that best divides the instances into their corresponding classes. Any change in dataset can lead to major changes in the model. Despite

having advantage over outliers and missing value, it has tendency to overfit the data. But this overfitting can be improved by limiting the tree depth. DT algorithm C4.5 can reduce the chances of overfitting but overfitting for noisy data may still happen. C5.0 is the updated version of C4.5, where the classifier is more accurate and much faster and less resource hungry [27]. C5.0 in combination with boosting algorithm can have increase the accuracy more.

For IDS the C5.0 classifier performs very well in terms of accuracy, detection and false alarm rate. More specifically it performs well in SID because SID uses a known signature database of attacks. As new attacks are detected, the data is stored in SID database, and it grows. Because C5.0 consumes a smaller number of resources and is much faster, it can be used to develop a SIDS with large amount of data in dataset/database.

2) *Support vector Machine (SVM):* SVM is ML technique used for classification, regression, and outlier detection problems. SVM generates lines/hyper-planes from the input samples and separates the data into classes for prediction [34]. SVM has different kernels including linear SVM and Non-linear SVM. In linear kernel, the SVM separates the training data using hyperplane. In non-linear kernel, if the training data is not linearly separable, the SVM maps the input data to high dimension space to find a linear hyperplane. SVM can be categorized into two types of classes including multi-class SVM and one-class SVM. Because of unsupervised ML capability of one-class SVM it is used for novel attack detection in AIDS. SVM is a complex algorithm and suffers from extensive resource requirement. But depending on the kernel and parameter, SVM performs differently. SVM usually is robust against noise, but the training time of SVM is very high. Linear SVM usually is less accurate and can be overfitting.

For IDS the one-class SVM classifier can perform well because of its unsupervised capability and good outlier detection capability. But it can be sensitive to training data and increase false alarm rate. IDS should be faster and efficient, but the training time is extremely high, and it is not desirable. SVM can be manipulated in diverse ways to improve these drawbacks, like splitting of data into smaller training sets, using higher value for parameter γ [31].

3) *Random Forest (RF):* RF is a ML technique used for regression and classification. RF utilizes bagging, an ensemble learning technique. It consists of a collection of decision trees and the greater the number of trees the greater the precision of the outcome [30]. It can also manage missing value and it does not require feature scaling. It works well with both categorical and continuous data. Like DT it is robust to outliers but comparatively it is less impacted by noise. RF reduces the overfitting issue of DT which also increases the precision. Unlike DT it is more complex and has longer training period.

For IDS the RF is a good choice because of its ability to perform good in noisy data. Network packets can have noisy data and RF can have good precision and low false detection rate compared to other algorithms. Network Packets have some features with continuous data which can be managed by RF and have better outcome. But because of the longer training period and complexity, it may require more resources which is not desirable of real time intrusion detection.

4) *Naïve Bayes (NB):* NB is based on Bayesian learning method. It is a probabilistic ML technique that uses probability (Bayes rule) to generate a probabilistic model. It works better with categorical value. For small training data, it can perform well than other classifiers. There are 3 NB algorithm including Gaussian NB, Bernoulli NB and Multinomial NB. Gaussian NB is used for continuous data values, Bernoulli NB is used for binary values and Multinominal NB is used for discrete values [34].

For IDS, NB is faster and simpler. However, it assumes that features are independent of each other, and this assumption may not hold true for detecting different types of attacks. In KDDCup99 dataset, features are highly dependent to each other, and this may not give desirable result when detecting diverse types of attacks [34]. Hidden Naïve Bayes is an extension of NB that relaxes this assumption.

5) *K-Means Clustering:* K-means algorithm is a clustering-based anomaly detection technique. It is based on the assumption that normal data lies close to their closest cluster and outlier lies far from their closest cluster. It can be both used for anomaly detection and refining dataset by separating the outliers from training data [34].

For IDS, K-means clustering can result in a good detection rate. But sometimes the normal data can be same as some attack features which can increase false positive rate. Sometime the anomaly data can form clusters itself and can increase the false negative rate. As it can detect outlier easily, it can be used for refining dataset for classifiers that are sensitive to outliers. This can increase the accuracy of other techniques when training with refined training dataset.

6) *K-Nearest Neighbor (KNN):* KNN is a simple yet effective instance-based learning algorithm [34]. It is a lazy learner because it delays generalization until classification. Because it is lazy, the efficiency is poor. It can train faster because it delays generalization until classification, but more computational power is required during classification. Depending on the value of "K", the accuracy differs on different train datasets. It classifies data based on the assumption that the test instance is close in proximity of other instances that have similar properties. Proximity is determined by the value of "K".

For IDS, KNN can result in good accuracy and detection rate depending on the value of "K". It can have low false rate but because of its poor efficiency it can result in higher classification time.

7) *Genetic Algorithm (GA):* GA is a search algorithm that finds a solution based on principles of natural selection and genetics [34]. The operators in GA are inspired by biological concepts. The operators are initialization, selection, crossover and mutation. In GA the initially selected population is evolved until the fittest genes survive. The fittest genes are found through iteration of these three processes: selection, crossover, mutation. GA is fast in building

classifiers and can handle noise efficiently [50]. GA can be used with other classifiers to optimize them. But GA does not assure of finding a global optimum and they need a large number of fitness function evaluation which increases complexity. Representing a problem space in GA is also complex.

For IDS, GA can increase the detection speed because it is fast in building classifiers. Because IDSs are adaptive system, retraining classifier with GA can also increase speed as well as accuracy. GA can be used with other classifier-based IDS to improve the detection rate and false negative rate.

8) *Fuzzy Logic:* Fuzzy logic is a technique that utilizes many-valued logic and deals with approximate reasoning rather than exact [34]. The logical decision is based on rule sets defined by if-then conditions. It is robust with imprecise input. It is easy to implement and flexible when editing the rules. So, it consumes less resource and requires less computational power. Fuzzy logic accuracy completely depends on the dataset used to model. Noise in training data can lead to greater inaccuracy. It requires a lot of testing and validation to fine tune the model. It is hard to develop a classifier with fuzzy logic than other ML technique because of complexity. Neuro Fuzzy logic is a technique where fuzzy logic is combined with Artificial Neural Network to increase the accuracy.

For IDS, Fuzzy logic alone is not enough to detect all types of attack. It usually works well when stacking with other classifiers. Because it depends on training dataset, small amount of attack data in database of IDS can cause low detection rate and high false negative rate. It can improve the detection rate and false alarm rate when using with ML algorithms.

9) *Artificial Neural Network (ANN):* ANN is a deep learning technique that uses three element including input node, hidden nodes, and output node [34]. It is modeled after the neurons of a brain. It is a non-linear model and easy to use technique. ANN represents problems by attribute-value pair. It is used for problems having target function, the output may be discrete-valued or real-valued. It is easy to implement and is usually fast when evaluating output. But because it uses parallel processing and is non-linear, it requires more computational power and takes longer time to train. It requires proper training to operate and is a flexible and adaptive technique. The larger the training dataset, the accurate it can get. Also, the more precise the feature set is, the more accurate it is. Some common ANNs are Back Propagation Algorithm, Recurrent Neural Network, Convolutional Neural Network, Radial basis function Neural Network, Feedforward Neural Network, etc.

For IDS, if the dataset is large, ANN can give a good detection capability. In case of multiple/collaborative attacks, ANN can lack because of its inability to restore to past events. Because of long training time, it can decrease the attack detection rate. Detecting small attacks means not every attack data stores in dataset, which can cause inability to detect new attacks.

10) *Ensemble Learning:* Ensemble learning is a technique of combining multiple learners and then combining the predictions made by base classifiers [34]. Ensemble learning provides a strong generalization capability compared to individual base classifiers. The base classifier is usually a machine learning algorithm such as DT, SVM, ANN, ND, K-means, etc. Random Forest is the most popular ensemble machine learning algorithm. This technique is more robust against noise. It has better accuracy than stand-alone classifiers. But combining classifiers can increase training time and requires more resources and computational power. There are many ways to combine the classifiers including bagging, boosting, and stacking [46]. Bagging is the simplest ensemble method for improving unstable classification problems. Bagging uses multiple versions of training set to train different models based on base classifier. The outputs of the models are combined to a single output by voting. Boosting is widely used to boost the performance of a set of weak classifiers. Boosting makes a weak classifier strong. Boosting provides sequential learning where first one learns the whole dataset and the following learns from the training sets based on the performance of the previous. AdaBoost is the most common boosting algorithm for weak classifiers. Stacking can be achieved by combining base classifiers as base learner and stacking model learner. Each of the trained base classifier's predicted output data is again learned by the combined classifier to make final prediction. The combination of the classifiers can give the best accuracy.

For IDS, ensemble approaches have the advantage that they can adapt to any change in network traffic [46]. Bagging technique can be useful when using intrusion datasets with large and high-dimensional data. Boosting can improve the detection rate and lower the false alarm rate. If the base classifier has initial high false alarm rate, it can decrease the detection rate and increase the false alarm rate by iterating the model with misclassified data. Stacking can improve the accuracy if the base classifiers give good prediction capability with the intrusion training dataset.

We have discussed some widely used techniques and their advantages and drawbacks for IDS. Some can detect network intrusion on its own. Some can give low false alarm rate and high detection rate which can be achieved by combing it with other techniques. Some have high training time, and some have fast prediction capability. Some can develop a refined dataset than can be used by other classifiers to give better outcome. Because each techniques have their pros and cons over developing an IDS, it is necessary to compare the best algorithms for high detection rate, high accuracy, low false alarm rate, less computational power, less resource hungry, faster training speed, and faster detection speed.

6. Comparative Analysis

There are different machine learning algorithms, discussed in previous section, which can be used as a comparative analysis for intrusion detection system. There are few intrusion detection datasets discussed previously. As most authors prefer to use NSL-KDD dataset, we will be comparing different machine learning algorithms for network intrusion detection using the NSL-KDD dataset discussed in [51], which improves on its predecessor KDD-99 but still inherits some of the drawbacks.

To develop a ML model, data needs to be preprocessed such as data cleaning, transforming, feature selection or reduction etcetera and then sampled to fix minority data issue. Data cleaning and transforming removes any outlier and then standardize the data, so that it can be used to develop model efficiently with better accuracy. Some of the features in NSL-KDD need preprocessing for some ML algorithms. Features including protocol type containing categorical data- TCP, UDP and ICMP needs to be transformed to binary data. Data sampling is necessary for NSL-KDD dataset which also has class population issue, where attacks type including R2L, U2R and Probe have low population, biasing ML model toward majority of the class and decreasing the accuracy and increasing false alarm rate. For training we have used "KDDTrain+" and for testing the model we have used "KDDTest+". There are 125973 instances in "KDDTrain+" and 22544 instances in "KDDTest+" with 41 features excluding the class feature in NSL-KDD shown in the Table 1.

Table 1. Features in NSL-KDD dataset

duration, protocol_type, service,flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in,num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate

For this comparative analysis scikit-learn [52] is used. Scikit-learn is a python-based machine learning library. To select the preprocessing model, we are going to compare different machine learning algorithms with different encoding, feature selection, scaling methods and then different sampling methods.

6.1. Performance Metrics

There are many performance matrices to use for comparing the performance of a machine learning algorithm. We will be comparing accuracy, precision, recall, train time and prediction time for our classification dataset.

1) *Accuracy:* Classification accuracy is the simplest metric to use and implement. It is the ratio of correct predictions to the total number of predictions. It is the percentage of corrected prediction for total number of predictions. For this analysis, the accuracy is the number of correctly predicted anomaly and normal data compared to the number of predictions made.

2) *Confusion Matrix:* Confusion Matrix is a tabular visualization of the ground-truth labels versus model predictions. Each row of the confusion matrix represents the instances in a predicted class and each column represents the instances in an actual class. Table 2 represents the confusion matrix of anomaly and normal data in this analysis. The true positive (TP) is the number of correctly predicted anomaly data and false positive (FP) is the number of incorrectly predicted anomaly data as normal data. The true negative (TN) is the number of correctly predicted normal data and false negative (FN) is the number of incorrectly predicted normal data as anomaly data. False positive (FP) and false negative (FN) is type I and type II error, respectively.

Table 2. Confusion Matrix

	Actually Anomaly	Actually Normal
Predicted anomaly	True Positive (TP)	False Positive (FP)
Predicted normal	False Negative (FN)	True Negative (TN)

In this analysis, as the amount of false prediction in Intrusion detection system is important for its performance, we will be comparing the false positive and false negative to evaluate the ML models. The lower the false rate the better the model is for an intrusion detection system.

3) *Precision:* Precision is the ratio of true positives and total positives predicted (1). The precision metric focuses on Type-I errors (FP). It cannot measure the existence of Type-II error, which is false negative. A precision score towards 1 will signify that the model did not miss any true positives, while a low score indicates a higher number of false positive. In this analysis, the precision is the ratio of correctly predicted anomaly data to total of correctly classified anomaly data and incorrectly classified anomaly data as normal data. A low score means high number of

incorrectly predicted normal as anomaly.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

4) *Recall:* A Recall is the ratio of true positives to all the positives in ground truth (2). The recall metric focuses on type-II errors (FN). It cannot measure the existence of Type-I error, which is false positive. A recall score toward 1 will signify that the model did not miss any true positive, while a low score indicates a higher number of false negative. In this analysis the recall is the ratio of correctly predicted anomaly data to total of correctly classified anomaly data and incorrectly classified normal data as anomaly data. A low sore means high number of incorrectly predicted anomaly as normal.

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

5) *Train time and Prediction time:* For this analysis, train time is the amount of time in seconds to train the whole model. Train time is important because, as new attack types are added to the database on every detection, the database will become large. So, a lower train time is desirable. The prediction time is the amount of time in seconds to predict all the 22544 instances in "KDDTest+" portion of the NSL-KDD dataset. Low prediction time is also desirable, for how fast a model can detect attacks.

6.2. Selected Machine Learning Algorithms

The previous section describes selected supervised machine learning algorithms that includes k-Nearest-Neighbors (KNN), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), Neural network (ANN), adaBoost etc. We are going to compare these machine learning algorithms on each step of data preprocessing steps to analyze which combination of algorithms are best to be fit for an intrusion detection system. We are not going to use SVM because of its complexity with large dataset and we are using "NSL-KDD" dataset with 125973 instances in "KDDTrain+" portion that is used for training the models in this comparative analysis. Because we are using a machine learning library named "Scikit Learn", the algorithm it uses for DT is an optimized version for CART. So, for DT we are going to use the CART algorithm with random state set to 0. For KNN we are going to use 10 nearest-neighbors as its parameter. For Random Forest, we are going to use 100 random states as its parameter. For Naïve Bayes, we are going to use its default parameters. For Neural network we are going to use Multilayer perceptron (MLP), that uses a backpropagation algorithm to train the model. For MLP, we are going to use (5,2) as hidden layer parameter and 10 as number of random states. AdaBoost is an ensemble boosting technique that uses multiple weak learners (small decision trees) and boost and combine them to make a strong learner. For AdaBoost we use 100 number of estimators and 0 number of random states. We compare each of these machine learning algorithms through data preprocessing steps.

6.3. Data Preprocessing

Data preprocessing is a key step in developing a machine learning model. Data preprocessing involves a lot of steps including encoding, feature scaling, feature reduction etc. For data preprocessing, we mapped the categorical feature data to binary data by applying one-hot encoding, which generates new features for each unique value. Then we compared few features scaling method for best performance of the model. Finally, we compared different feature reduction techniques for best performance of the model.

1) *Encoding:* Many machine learning algorithms give inferior performance or unexpected result when learning with categorical data. They give more significance to higher numbers and train the model biased toward that category. So, we need to convert the categorical features to numeric/binary features. For the NSL-KDD we applied one-hot encoding to the categorical features including protocol type, service, and flag. One-hot encoding converts each categorical feature into new features and assign a binary value of 1 or 0 to those features. For example: For an instance in NSL-KDD with protocol_type "TCP", one-hot encoding converts the "protocol_type" feature to "protocol_type_icmp", "protocol_type_tcp" and "protocol_type_udp" and then assigns value "1" to equivalent column that represents TCP type which is "protocol_type_tcp" and assigns "0" to the rest. The Table 3 demonstrates the feature change in protocol-type, service, and flag after applying one-hot encoding. The total number of features in "NSL-KDD" feature set transforms from 41 features to 122 features. The features changed in the above table are the only changes, rest of the features remained same.

2) *Feature scaling:* Feature scaling is a process used to normalize the range of independent variables or features of data to a particular range or scale. Feature scaling helps in speeding up the calculations in deep learning algorithms and majority of machine learning algorithms. They perform much better when dealing with features that are on the same scale. Difference in scale across the features can slow down the algorithm as well as reduce accuracy. This inferior performance is due to sensitivity to input values during learning, which can result in higher generalization error. "NSL-KDD" dataset does not follow normal distribution and because each feature has range of values in different scales. We

will be applying standardization and normalization individually to analyze the outcome of these feature scaling techniques on our dataset.

Normalization is a technique applied in data preprocessing for machine learning to change the values of numeric columns to a common scale, without losing information. Normalization rescales the columns with numeric values into a range of [0,1]. Normalization helps solve biasing problem of larger values for some ML algorithms. Standardization is a similar technique applied in during data preprocessing. Standardization rescales the columns with numeric values to have a mean of 0 and a standard deviation of 1.

Normalization is effective when the distribution of the dataset does not follow a Gaussian/Normal distribution which can be effective when using KNN, SVM or K-Means as models. Standardization can help when the data follows Gaussian/Normal distribution. Decision Tree and Random Forest algorithms build models using information and Naïve Bayes uses probability-based algorithm to train the model. Tree-based algorithms and probability-based algorithms are insensitive to the scale of the features. So, none of DT, RF and NB needs normalized or standardized data to train the model. For the sake of comparison, we are still going to compare each selected machine learning algorithm with these feature scaling techniques.

Table 3. Transformed features of NSL-KDD

Before applying One-Hot Encoding	After applying One-Hot Encoding
protocol_type, service, flag	protocol_type_icmp, protocol_type_tcp, protocol_type_udp, service_IRC, service_X11, service_Z39_50, service_aol, service_auth, service_bgp, service_courier, service_csnet_ns, service_ctf, service_daytime, service_discard, service_domain, service_domain_u, service_echo, service_eco_i, service_ecr_i, service_efs, service_exec, service_finger, service_ftp, service_ftp_data, service_gopher, service_harvest, service_hostnames, service_http, service_http_2784, service_http_443, service_http_8001, service_imap4, service_iso_tsap, service_klogin, service_kshell, service_ldap, service_link, service_login, service_mtp, service_name, service_netbios_dgm, service_netbios_ns, service_netbios_ssn, service_netstat, service_nnsp, service_nntp, service_ntp_u, service_other, service_pm_dump, service_pop_2, service_pop_3, service_printer, service_private, service_red_i, service_remote_job, service_rje, service_shell, service_smtp, service_sql_net, service_ssh, service_sunrpc, service_supdup, service_systat, service_telnet, service_tftp_u, service_tim_i, service_time, service_urh_i, service_urp_i, service_uucp, service_uucp_path, service_vmnet, service_whois, flag_OTH, flag_REJ, flag_RSTO, flag_RSTOS0, flag_RSTR, flag_S0, flag_S1, flag_S2, flag_S3, flag_SF, flag_SH

After applying standardization and normalization to the encoded dataset individually, we applied each selected machine learning algorithm to test the outcome compared to an unscaled model. Table 4 shows the outcome of this comparison of feature scaling techniques on different machine learning algorithms. The Following points (i – vi) are the interpretations for the comparative analysis of this sub section.

i) For KNN, normalization increased the accuracy, precision and recall for an increase in train time and prediction time. Using standardization did not have much improvement on the performance.

ii) For DT, standardization improved the accuracy and recall but decreased the precision while maintaining a similar train time and prediction time compared to the default DT model without any feature scaling technique. Normalization on the other hand decreased the accuracy of the model with greater increase in train time.

iii) For Random Forest none of the feature scaling techniques improved any performance of the model.

iv) For NB, the model had the worst performance among all the ML algorithms in this comparison and though normalization improved the accuracy of the model, it still has the worst performance. When using standardization, the recall of the model was better, but when using normalization, the precision of the model was better.

v) For MLP, standardization increased the accuracy, precision and recall while having similar train time and prediction time compared to the default MLP model without any feature reduction technique. Normalization decreased the performance of the model overall.

vi) For adaBoost, standardization had the most increase in accuracy and recall, with a slight decrease in precision compared to the default adaBoost model without any feature reduction technique. The train time and prediction time was also similar. Among all the comparison standardization with adaBoost had the best accuracy among all the ML algorithms

For intrusion detection system, the precision and recall are more important because a lower amount of type I and II errors (FP and FN) is desirable. So, for further analysis we are going to use the following (i-v) as default models.

i) KNN + Normalization
ii) DT + standardization
iii) RF
iv) MLP + standardization
v) AdaBoost + standardization

Because of the inferior performance of Naïve Bayes (NB) model, we are not going to compare it further in our comparative analysis for an intrusion detection system.

3) *Dimensionality/Feature reduction:* Feature reduction, also known as dimensionality reduction, is a process used to reduce the number of features in a large dataset. Reducing the number of features that does not have much impact on the model can lead to a lower computational power and faster train time and prediction time but provide negligible or no increase in prediction accuracy [53]. For this analysis we will be comparing different feature reduction techniques by using it on the model of methods we have selected until previous subsection. There are different feature reduction techniques including low variance filter, high correlation filter, Random Forest, Incremental Principal Component Analysis (Incremental PCA) etc.

Correlation filter filters out all the columns that have a high correlation with other columns above a certain threshold. The correlation filter calculates the correlation between each feature and drops the features that has a threshold above certain percentage. Low variance filter filters out the features that have low variance below a certain threshold. The features that impact the model less is said to have low variance. Like Correlation filter, low variance filter can improve the performance of a model. Random forest is a classifier but can also be used as a feature selection/reduction technique. Incremental PCA is like PCA, it is used for large dataset to fit in memory easily, it uses batch size to control memory and like PCA it can select the top principal components according to own choice.

Table 4. Comparing feature scaling techniques

Algorithms	**Accuracy**	**Precision**	**Recall**	**Train time**	**Prediction time**
K-Nearest Neighbors (KNN)					
KNN only	0.7736	0.9657	0.6246	0.35466	48.16329
KNN + Standardization	0.7803	0.9230	0.6698	0.64645	52.11184
KNN + Normalization	0.8087	0.9689	0.6860	0.52629	54.47347
Decision Tree (CART)					
DT only	0.8118	0.9677	0.6926	1.97002	0.01564
DT + Standardization	0.8176	0.8064	0.8943	2.26852	0.01075
DT + Normalization	0.7922	0.9688	0.6560	6.13145	0.01074
Random Forest (RF)					
RF alone	0.7818	0.9687	0.6373	11.98722	0.31595
RF + Standardization	0.7548	0.9007	0.6398	11.64288	0.17479
RF+ Normalization	0.7746	0.9691	0.6239	23.54962	0.26256
Naïve Bayes (NB)					
NB only	0.4503	0.9366	0.0369	0.56055	0.04883
NB + Standardization	0.5694	0.5694	0.9991	1.01705	0.04493
NB + Normalization	0.5265	0.9671	0.1742	0.82742	0.04545
Neural Network- Multi Layer Perceptron (MLP)					
MLP only	0.7742	0.9025	0.6765	20.52753	0.01758
MLP + Standardization	0.8181	0.9190	0.7462	22.22811	0.01171
MLP + Normalization	0.7622	0.9159	0.6412	64.05987	0.00976
AdaBoost					
AdaBoost only	0.7966	0.9267	0.6979	27.89584	0.78510
AdaBoost + Standardization	0.8273	0.8944	0.7899	29.62245	0.80352
AdaBoost + Normalization	0.7773	0.9239	0.6634	46.69595	0.92091

After individually applying Correlation Filter, Low Variance Filter, Random Forest and Incremental PCA to the encoded and scaled dataset, we applied each selected machine learning algorithm to test the outcome. For Correlation Filter and Low Variance filter we used a threshold of 0.9, which drops features if the correlation between features is greater than 90% and if variance of data is less than 90%. For random forest we used 100 estimators (number of trees) to reduce feature set. For Incremental PCA, it selects the top uncorrelated features, so we selected the top 50 features/components.

Some ML algorithms tend to perform poor when features are highly correlated or when features have low variance. After applying the correlation filter the feature set was reduced to 108 features for each model, which means the rest 14 features have high correlation and will not improve on the model's accuracy. After applying the low variance filter, we got only 3 features selected for models with normalization applied and 115 features with standardization applied. Because it is not ideal to classify data or develop intrusion detection system with such small number of features, the low variance filter is not ideal for this model. After applying random forest 25 features were selected, but because of its low performance in accuracy, even though it had much less train time, the technique is not ideal of an intrusion detection system. After applying PCA and selecting the top 50 features/components, the accuracy decreases, but the train time is lower than average train time in this comparison. As we need better accuracy, precision, recall with lower train time and prediction time, the correlation filter is best fit for our model.

Table 5 shows the outcome of this comparison of feature reduction techniques on different models of different machine learning algorithm selected from previous subsection. The Following points (i – iv) are the interpretations for the comparative analysis of this section sub section.

i) For "KNN + normalization", correlation filter increased the accuracy and recall slightly. But the train time increased by about 6 times while the prediction time was similar. Each of the other feature reduction techniques decreased the performance of this model.

ii) For "DT + standardization," each of the feature reduction techniques excluding PCA, had similar performance of accuracy, precision and recall. But among them the model with correlation filter had the most accuracy, precision and with a train time double than the default model and lower prediction time. PCA had the most decrease in performance of the Decision Tree with train time that is very undesirable.

iii) Like "DT + standardization", the performance of RF increased the most after applying the correlation filter with a slight increase in train time and decrease in prediction time. PCA also decreased the performance of the RF model.

iv) For "MLP + standardization", the correlation filter increased the accuracy and precision slightly, but the train time increased a lot, almost twice the default train time.

v) For "AdaBoost + standardization", the original model had the highest accuracy, precision and recall among all the feature reduction techniques with the lowest train time.

Table 5. Comparing feature reduction techniques

Algorithms	**Accuracy**	**Precision**	**Recall**	**Train time**	**Prediction time**
KNN + Normalization					
Default	0.8087	0.9689	0.6860	0.52629	54.47347
with Correlation Filter	0.8111	0.9680	0.6910	3.24214	56.26599
with Low Variance Filter	0.7944	0.9631	0.6644	0.92604	0.52292
with Random Forest	0.7789	0.9660	0.6339	24.27237	53.57079
with Incremental PCA	0.7601	0.8872	0.6628	80.36627	56.11899
DT + standardization					
Default	0.8176	0.8064	0.8943	2.26852	0.01075
with Correlation Filter	0.8210	0.8072	0.9006	4.75919	0.00782
with Low Variance Filter	0.8182	0.8024	0.9031	2.41615	0.01171
with Random Forest	0.8144	0.7980	0.9024	18.79779	0.00390
with Incremental PCA	0.6375	0.7553	0.5373	94.82595	0.00684
RF					
Default	0.7818	0.9687	0.6373	11.98722	0.31595
with Correlation Filter	0.7901	0.9688	0.6523	14.18187	0.22948
with Low Variance Filter	0.7644	0.9660	0.6075	10.72580	0.24225
with Random Forest	0.7686	0.9672	0.6144	25.39306	0.22948
with Incremental PCA	0.4320	0.7843	0.0031	201.85706	0.22850
MLP + standardization					
Default	0.8181	0.9190	0.7462	22.22811	0.01171
with Correlation Filter	0.8193	0.9218	0.7458	47.84748	0.00977
with Low Variance Filter	0.7957	0.9020	0.7192	34.64633	0.01269
with Random Forest	0.7480	0.9141	0.6152	30.57032	0.00781
with Incremental PCA	0.7467	0.8078	0.7283	111.85413	0.00684
AdaBoost + standardization					
Default	0.8273	0.8944	0.7899	29.62245	0.80352
with Correlation Filter	0.8208	0.8834	0.7894	31.92090	0.67965
with Low Variance Filter	0.8273	0.8944	0.7899	33.18726	0.88569
with Random Forest	0.7680	0.8826	0.6833	33.44512	0.35545
with Incremental PCA	0.7425	0.8422	0.6740	193.38850	0.47458

Correlation Filter removes the features that are less correlated and that is ideal for an intrusion detection system, because in a packet in network traffic, the variables that classify traffic as anomaly or normal, are correlated to each other. In this comparison, the correlation filter improved the performance of every machine learning model except for AdaBoost. So, because correlation filter performed better with our models for "NSL-KDD" dataset, we will be using this feature reduction technique on each model except AdaBoost to continue the comparative analysis. For further analysis we are going to use the following (i-v) as default models.

i) KNN + Normalization + correlation filter
ii) DT + standardization + correlation filter
iii) RF + correlation filter
iv) MLP + standardization + correlation filter
v) AdaBoost + standardization

6.4. Sampling

Some machine learning algorithms give inferior performance due to imbalance in class distribution in the training dataset. Many machine learning algorithms give best performance when there is equal number of observations for each class. When there are small numbers of instances for a specific class, the algorithm tends to model the classifier biased

towards the class with large number of instances. So, to solve this imbalance problem we do sampling. Sampling is a technique that whether by over-sampling the minority class or by under-sampling the majority class, balances the class distribution of the training set by duplicating instances. It makes duplication of instances in the same range. Because there are several types of classification problems, no single sampling method is best. For our dataset "NSL-KDD" we have an imbalance problem and because it has fewer instances of few attack types, we are going to apply few over-sampling techniques including SMOTE, Borderline-SMOTE, ADASYN. We have chosen to use over-sampling technique because, our dataset has crucial data about attack types and removing instances can lead to inferior performance for the intrusion detection system. But we must be careful because sampling may lead to overfitting of data that can lead to lower accuracy, precision, and recall. Sampling can be done anytime during building the model, but the dataset needs to be encoded first.

Synthetic Minority Oversampling Technique (SMOTE) [54] is a data augmenting technique that duplicates data of minority class and synthesizes new instances/samples from existing dataset. For this analysis we have selected SMOTE with 10 number of k-nearest neighbors to synthesize instances. Borderline-SMOTE [55] is a variant of the original SMOTE, but instead it detects the borderline samples and synthesized new instances/samples from existing dataset. For this analysis we used borderline SMOTE of kind "borderline-2" with 10 numbers of k-nearest neighbors. ADASYN [56] is also like SMOTE but instead of generating samples of instances for all classes, it generates instances inversely proportional to the density of minority class instances. It generates more synthetic samples in region of the feature space where the density of minority instances is low. For this analysis we used ADASYN with n-nearest neighbors of 10.

Table 6 show the outcome of this comparison of different over-sampling techniques on different models of different machine learning algorithm selected from previous subsection. Because sampling is a technique that randomly generates instances to solve imbalance problems, the outcome is slightly different each time. So, we took the last outcome from the analysis. The train time and prediction time also taken without considering the time required for sampling the dataset. As expected, the train time and prediction time would have been higher than usual if the sampling time were considered. For this comparison we will be comparing only the train and prediction time regardless the over-sampling technique and the time required for over-sampling. The Following points (i –v) are the interpretations for the comparative analysis of this section sub section.

i) For our first model, "KNN + Normalization + correlation filter", using Borderline SMOTE as oversampling technique, improved the performance of the model with similar train time and slightly greater prediction time. After analyzing it multiple times for comparison, no overfitting problem arose.

ii) For "DT + standardization + correlation filter", using SMOTE improved the accuracy and recall sightly with a slight increase in train time and prediction time. But during the analysis, each oversampling technique was overfitting the data two out of three times, resulting in a drastic decrement in performance on each analysis.

iii) For "RF + correlation filter", using Borderline SMOTE improved the performance of model with slightly greater train time and prediction time. Using SMOTE overfitted the data resulting in a slight decrease in performance.

iv) For "MLP + standardization + correlation filter", using Borderline SMOTE improved the performance of the model with lower train time and same prediction time. But each oversampling technique overfitted the data one out of three times during analysis.

v) For "AdaBoost + standardization", because of its similarity with DT, only SMOTE had improvement on the performance of the model. Other oversampling technique overfitted the data resulting in inferior performance.

Borderline SMOTE improved the accuracy for models with KNN, RF, MLP but overfitted the data for DT, AdaBoost. On the other hand, SMOTE improved the performance of DT and Adaboost because other oversampling techniques were overfitting the data. So, because of overfitting the data, we need to carefully choose which oversampling technique to use to improve the performance of the mode. Lastly for this comparative analysis we are going to choose the best model from the following (i-v) points.

i) KNN + Normalization + correlation filter + Borderline SMOTE
ii) DT + standardization + correlation filter + SMOTE
iii) RF alone + correlation filter + Borderline SMOTE
iv) MLP + standardization + correlation filter + Borderline SMOTE
v) AdaBoost + standardization + SMOTE

Table 7 shows the best model for each ML selected after the comparative analysis. Among these selected models, "KNN + Normalization + correlation filter" with Borderline SMOTE had the best accuracy with high precision and low train time, but greater prediction time. "MLP + standardization + correlation filter" with Borderline SMOTE has the 2nd best accuracy with similar precision to KNN and greater train time, but lower prediction time. Lastly, "DT + standardization + correlation filter" and "AdaBoost + standardization" with SMOTE had great accuracy and the best recall among the ML models in this comparison but the precision is much lower. The train time and prediction time on the other hand was very less for the model with DT. The next paragraph explains the results of this comparative analysis.

The focus of the comparative analysis is to have the highest accuracy, precision and recall with low train time and prediction time. Each model had its drawbacks and lacking. After selecting the better combinations of each machine learning algorithms with feature scaling, feature reduction and over-sampling technique, we have found that for accuracy, KNN gave the best performance of 85.3% but the prediction time is comparatively large with 62.54 seconds. MLP had the second-best accuracy with 84.76% but the train time is comparatively very larger with 39.35 seconds. For low train time and prediction time, DT gave the best performance of 5.2 and 0.01 seconds respectively, which can be idea for IDS with heavy traffic flow. Random Forest, though having best precision of 96.89%, it had the lowest accuracy of 81.54%. So, for overall best performance considering accuracy, KNN with 85.3% accuracy will be best for IDS.

Table 6. Comparing over-sampling techniques

Algorithms	**Accuracy**	**Precision**	**Recall**	**Train time**	**Prediction time**
KNN + Normalization + correlation filter					
Default	0.8111	0.9680	0.6910	3.24214	56.26599
with SMOTE	0.8125	0.9671	0.6943	3.37844	62.32286
with Borderline SMOTE (Borderline-2)	0.8530	0.9620	0.7721	3.38974	62.53617
with ADASYN	0.8419	0.9604	0.7534	3.14338	54.64637
DT + standardization + correlation filter					
Default	0.8210	0.8072	0.9006	4.75919	0.00782
with SMOTE	0.8242	0.8031	0.9158	5.20487	0.01173
with Borderline SMOTE (Borderline-2)	0.6493	0.6512	0.8269	5.40624	0.01075
with ADASYN	0.8116	0.8405	0.8256	4.79522	0.00782
RF + correlation filter					
Default	0.7901	0.9688	0.6523	14.18187	0.22948
with SMOTE	0.7853	0.9687	0.6436	14.21274	0.21646
with Borderline SMOTE (Borderline-2)	0.8154	0.9689	0.6981	23.09821	0.26188
with ADASYN	0.8118	0.9699	0.6909	16.88498	0.38499
MLP + standardization + correlation filter					
Default	0.8193	0.9218	0.7458	47.84748	0.00977
with SMOTE	0.7981	0.8962	0.7298	35.37569	0.01563
with Borderline SMOTE (Borderline-2)	0.8476	0.9583	0.7657	39.35416	0.00782
with ADASYN	0.8086	0.9102	0.7364	34.90677	0.01075
AdaBoost + standardization					
Default	0.8273	0.8944	0.7899	29.62245	0.80352
with SMOTE	0.8336	0.8820	0.8170	30.17995	0.71429
with Borderline SMOTE (Borderline-2)	0.6064	0.5912	0.9998	34.49567	0.86609
with ADASYN	0.6952	0.6514	0.9994	32.06765	0.81631

Table 7. Selected ML techniques performance comparison

Algorithms	**Accuracy**	**Precision**	**Recall**	**Train time**	**Prediction time**
KNN + Normalization + correlation filter + Borderline SMOTE	0.8530	0.9620	0.7721	3.38974	62.53617
DT + standardization + correlation filter + SMOTE	0.8242	0.8031	0.9158	5.20487	0.01173
RF + correlation filter + Borderline SMOTE	0.8154	0.9689	0.6981	23.09821	0.26188
MLP + standardization + correlation filter + Borderline SMOTE	0.8476	0.9583	0.7657	39.35416	0.00782
AdaBoost + standardization + SMOTE	0.8336	0.8820	0.8170	30.17995	0.71429

7. Conclusion and Future Works

Network intrusion affects the services available through the internet since intrusions affects the CIA triad of information assurance (confidentiality, integrity, and availability). Such intrusions compromise online services and data that costs millions of dollars. Intrusion Detection Systems (IDS) is a measure against such network intrusion. The Automation of IDS reduces human intervention and accuracy of IDS's intrusion/attack detection makes it more reliable with respect to other measures against intrusions. Application of machine learning techniques in IDS can improve its accuracy. In this paper, the best machine learning algorithm, among the popular ones, is determined for a popular cyber security dataset (NSL-KDD). This research of finding the best machine learning algorithm is challenged by the vast number of feature set of the dataset. The larger the number of features, the higher the computation cost to work with the dataset. Reducing these features, hence, can help education the computation cost. Hence, the achievement of any study related to machine learning algorithms application on IDS data shall be the selection of appropriate machine learning algorithm and appropriate set of features. Therefore, our goal was to investigate and compare the machine learning

techniques with different feature scaling, feature reduction and over sampling technique and determine the best combination of techniques with optimum performance for intrusion detection systems.

Our research investigates machine learning algorithms that include Decision Tree, Support Vector Machine, Random Forest, Naïve Bayes, Neural network, and adaBoost. Our research was broken down into 4 steps. The first step is encoding the dataset. The second step is investigating and selecting the better pairs among each machine learning algorithm and different feature scaling technique pairs. The third step is investigating and selecting the better pairs among each of the selected pair from step two and different feature reduction technique. The fourth step is investigating and selecting the better pairs among each of the selected pair from step three and different over-sampling technique. Among the pairs selected in step four, KNN fulfilled the criteria of having higher accuracy, low false positive and low training time with respect to other pair of techniques in this paper.

As a future research direction, this study can be extended by implementing the models in a sandbox environment, where these models will be implemented in an IDS to test real-time scenarios. Additional machine learning techniques, e.g., deep learning can also be implemented in IDSs. Further, different deep learning techniques can be compared for intrusion detection and the best of them can be tested in a sandbox environment.

References

[1] G. Padmavathi, S. Divya, A Survey on Various Security Threats and Classification of Malware Attacks , Vulnerabilities and Detection Techniques, (2013).

[2] J. Andress, S. Winterfeld, The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice: Second Edition, The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice: Second Edition. (2014) 1–217.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in Cloud, Undefined. 36 (2013) 42–57. https://doi.org/10.1016/J.JNCA.2012.05.003.

[4] A. Mallik, A. Ahsan, M.M.Z. Shahadat, J.C. Tsou, Man-in-the-middle-attack: Understanding in simple words, International Journal of Data and Network Science. 3 (2019) 77–92. https://doi.org/10.5267/J.IJDNS.2019.1.001.

[5] R. v. Deshmukh, K.K. Devadkar, Understanding DDoS Attack & its Effect in Cloud Environment, Procedia Computer Science. 49 (2015) 202–210. https://doi.org/10.1016/J.PROCS.2015.04.245.

[6] E.M. Rudd, A. Rozsa, M. Günther, T.E. Boult, A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions, IEEE Communications Surveys and Tutorials. 19 (2017) 1145–1172. https://doi.org/10.1109/COMST.2016.2636078.

[7] S. Kamiya, J.-K. Kang, J. Kim, A. Milidonis, R.M. Stulz, What is the Impact of Successful Cyberattacks on Target Firms?, NBER Working Papers. (2018). https://ideas.repec.org/p/nbr/nberwo/24409.html (accessed January 16, 2022).

[8] M. Bada, J.R.C. Nurse, The social and psychological impact of cyberattacks, Emerging Cyber Threats and Cognitive Vulnerabilities. (2020) 73–92. https://doi.org/10.1016/B978-0-12-816203-3.00004-6.

[9] Y.Y. Wee, W.P. Cheah, S.C. Tan, K. Wee, Reasoning with Cause and Effect in Intrusion Detection, International Journal of Computer and Electrical Engineering. (2012) 641–646. https://doi.org/10.7763/IJCEE.2012.V4.574.

[10] B. Mukherjee, T.L. Heberlein, K. Levitt, Network intrusion detection, Undefined. 8 (1994) 26–41. https://doi.org/10.1109/65.283931.

[11] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity. 2 (2019) 1–22. https://doi.org/10.1186/S42400-019-0038-7/FIGURES/8.

[12] R. Patgiri, U. Varshney, T. Akutota, R. Kunde, An Investigation on Intrusion Detection System Using Machine Learning, Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018. (2019) 1684–1691. https://doi.org/10.1109/SSCI.2018.8628676.

[13] V. Jyothsna, V. v. Rama Prasad, K. Munivara Prasad, A Review of Anomaly based Intrusion Detection Systems, International Journal of Computer Applications. 28 (2011) 26–35. https://doi.org/10.5120/3399-4730.

[14] S. Einy, C. Oz, Y.D. Navaei, The Anomaly- And Signature-Based IDS for Network Security Using Hybrid Inference Systems, Mathematical Problems in Engineering. 2021 (2021). https://doi.org/10.1155/2021/6639714.

[15] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers and Security. 28 (2009) 18–28. https://doi.org/10.1016/J.COSE.2008.08.003.

[16] T. Verwoerd, R. Hunt, Intrusion detection techniques and approaches, Computer Communications. 25 (2002) 1356–1365. https://doi.org/10.1016/S0140-3664(02)00037-3.

[17] C. Kruegel, T. Toth, Using Decision Trees to Improve Signature-Based Intrusion Detection, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2820 (2003) 173–191. https://doi.org/10.1007/978-3-540-45248-5_10.

[18] M.K. Asif, T.A. Khan, T.A. Taj, U. Naeem, S. Yakoob, Network Intrusion Detection and its strategic importance, BEIAC 2013 - 2013 IEEE Business Engineering and Industrial Applications Colloquium. (2013) 140–144. https://doi.org/10.1109/BEIAC.2013.6560100.

[19] S.K. Gautam, H. Om, Computational neural network regression model for Host based Intrusion Detection System, Perspectives in Science. 8 (2016) 93–95. https://doi.org/10.1016/J.PISC.2016.04.005.

[20] P. de Boer, P. de Boer, M. Pels, Host-based Intrusion Detection Systems, (2005). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.2707 (accessed January 16, 2022).

[21] Bace, R. Gurley. (1999). Intrusion detection / Rebecca Gurley Bace. In *Intrusion detection*. Macmillan Technical Publishing.

[22] O. Depren, M. Topallar, E. Anarim, M.K. Ciliz, An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, Expert Systems with Applications. 29 (2005) 713–722. https://doi.org/10.1016/J.ESWA.2005.05.002.
[23] N. Das, T.S.-I.J. of A.N. and, undefined 2014, Survey on host and network based intrusion detection system, Academia.Edu. (n.d.). https://www.academia.edu/download/43954388/Survey_on_Host_and_Network_Based_Intrusi20160321-26062-7stpvd.pdf (accessed January 16, 2022).
[24] R. Brackney, Cyber-intrusion response, Proceedings of the IEEE Symposium on Reliable Distributed Systems. (1998) 413–415. https://doi.org/10.1109/RELDIS.1998.740533.
[25] S. Jose, D. Malathi, B. Reddy, D. Jayaseeli, A Survey on Anomaly Based Host Intrusion Detection System, Journal of Physics: Conference Series. 1000 (2018) 012049. https://doi.org/10.1088/1742-6596/1000/1/012049.
[26] Y. Shin, K. Kim, Comparison of anomaly detection accuracy of host-based intrusion detection systems based on different machine learning algorithms, International Journal of Advanced Computer Science and Applications. (2020) 252–259. https://doi.org/10.14569/IJACSA.2020.0110233.
[27] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, A. Alazab, Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine, Electronics (Switzerland). 9 (2020). https://doi.org/10.3390/ELECTRONICS9010173.
[28] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, A. Alazab, A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks, Electronics 2019, Vol. 8, Page 1210. 8 (2019) 1210. https://doi.org/10.3390/ELECTRONICS8111210.
[29] M.A. Aydin, A.H. Zaim, K.G. Ceylan, A hybrid intrusion detection system design for computer network security, Undefined. 35 (2009) 517–526. https://doi.org/10.1016/J.COMPELECENG.2008.12.005.
[30] R.M. Elbasiony, E.A. Sallam, T.E. Eltobely, M.M. Fahmy, A hybrid network intrusion detection framework based on random forests and weighted k-means, Ain Shams Engineering Journal. 4 (2013) 753–762. https://doi.org/10.1016/J.ASEJ.2013.01.003.
[31] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Systems with Applications. 41 (2014) 1690–1700. https://doi.org/10.1016/J.ESWA.2013.08.066.
[32] Z. Chiba, N. Abghour, K. Moussaid, A. el Omri, M. Rida, A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network, Procedia Computer Science. 83 (2016) 1200–1206. https://doi.org/10.1016/J.PROCS.2016.04.249.
[33] A.K. Dalai, S.K. Jena, Hybrid network intrusion detection systems: A decade's perspective, Lecture Notes in Electrical Engineering. 395 (2017) 341–349. https://doi.org/10.1007/978-81-322-3592-7_35.
[34] P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection, Undefined. 21 (2019) 686–728. https://doi.org/10.1109/COMST.2018.2847722.
[35] N. Moustafa, J. Slay, The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems, (2017) 25–31. https://doi.org/10.1109/BADGERS.2015.014.
[36] S. Sapre, P. Ahmadi, K. Islam, A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets Through Various Machine Learning Algorithms, (2019). https://arxiv.org/abs/1912.13204v1 (accessed January 16, 2022).
[37] M. Mohammed, M.B. Khan, E.B.M. Bashie, Machine learning: Algorithms and applications, Machine Learning: Algorithms and Applications. (2016) 1–204. https://doi.org/10.1201/9781315371658.
[38] B.M.-I.J. of S. and R. (IJSR), undefined 2020, Machine Learning Algorithms-A Review, Researchgate.Net. (2018). https://doi.org/10.21275/ART20203995.
[39] I. Muhammad, Z. Yan, SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY, Undefined. 05 (2015) 946–952. https://doi.org/10.21917/IJSC.2015.0133.
[40] N. Williams, S. Zander, G. Armitage, A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification, Computer Communication Review. 36 (2006) 7–15. https://doi.org/10.1145/1163593.1163596.
[41] S. Choudhury, A. Bhowal, Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection, Undefined. (2015) 89–95. https://doi.org/10.1109/ICSTM.2015.7225395.
[42] K.A. Jalil, M.H. Kamarudin, M.N. Masrek, Comparison of Machine Learning algorithms performance in detecting network intrusion, Undefined. (2010) 221–226. https://doi.org/10.1109/ICNIT.2010.5508526.
[43] B. Dong, X. Wang, Comparison deep learning method to traditional methods using for network intrusion detection, Undefined. (2016) 581–585. https://doi.org/10.1109/ICCSN.2016.7586590.
[44] M. Almseidin, M. Alzubi, S. Kovacs, M. Alkasassbeh, Evaluation of Machine Learning Algorithms for Intrusion Detection System, SISY 2017 - IEEE 15th International Symposium on Intelligent Systems and Informatics, Proceedings. (2018) 277–282. https://doi.org/10.1109/SISY.2017.8080566.
[45] H. Malhotra, P.S.-I.J. of Computer, undefined 2019, Intrusion Detection using Machine Learning and Feature Selection., J.Mecs-Press.Net. (n.d.). http://j.mecs-press.net/ijcnis/ijcnis-v11-n4/IJCNIS-V11-N4-6.pdf (accessed January 27, 2022).
[46] I. Syarif, E. Zaluska, A. Prugel-Bennett, G. Wills, Application of Bagging, Boosting and Stacking to Intrusion Detection, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 7376 LNAI (2012) 593–602. https://doi.org/10.1007/978-3-642-31537-4_46.
[47] K. Atefi, S. Yahya, A.Y. Dak, A. Atefi, A hybrid intrusion detection system based on different machine learning algorithms, Undefined. (2013).
[48] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, H. Jingjing, Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms, Security and Communication Networks. 2019 (2019). https://doi.org/10.1155/2019/7130868.
[49] A. Iqbal, S. Aftab, A Feed-Forward and Pattern Recognition ANN Model for Network Intrusion Detection, International Journal of Computer Network and Information Security. 11 (2019) 19–25. https://doi.org/10.5815/IJCNIS.2019.04.03.

[50] P.G. Majeed, S. Kumar, Genetic Algorithms in Intrusion Detection Systems: A Survey, International Journal of Innovation and Applied Studies. 5 (2014) 233–240. http://www.ijias.issr-journals.org/abstract.php?article=IJIAS-13-284-07 (accessed January 16, 2022).
[51] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009. (2009). https://doi.org/10.1109/CISDA.2009.5356528.
[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research. 12 (2012) 2825–2830. https://arxiv.org/abs/1201.0490v4 (accessed January 16, 2022).
[53] Ahsan M, Gomes R, Chowdhury MM, Nygard KE. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. Journal of Cybersecurity and Privacy. 2021; 1(1):199-218. https://doi.org/10.3390/jcp1010011
[54] N. v. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research. 16 (2002) 321–357. https://doi.org/10.1613/JAIR.953.
[55] H. Han, W.Y. Wang, B.H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 3644 LNCS (2005) 878–887. https://doi.org/10.1007/11538059_91.
[56] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, Proceedings of the International Joint Conference on Neural Networks. (2008) 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969.

Authors' Profiles

Shadman Latif is born in Bangladesh on 25th December 1999. He did his BSc in Computer Science and Engineering from American International University-Bangladesh in 2022. His research focus is on the applications of Machine Learning Algorithms and Artificial Intelligence. His future target is to achieve a prestigious Masters and PhD degree in the field of Machine Learning and Artificial Intelligence.

Faria Farzana Dola is born in Bangladesh on 24th September 1999. Currently she is pursuing her Bachelor's degree in Computer Science and Engineering under the Faculty of Science and Technology at American International University- Bangladesh.

MD. Mahir Afsar is an undergraduate student of the computer science department at the American International University-Bangladesh. His general research interest is in the area of intrusion detection and machine learning algorithms.

Ishrat Jahan Esha born in Bangladesh on 18th Octobar 2000. She is a student of American International University of Bangladesh. Her general research interest is in the area of information technology and machine learning algorithms. Her future target is to achive Masters degree in the field of machine learning.

Prof. Dr. Dip Nandi is the Director of Faculty of Science & Technology (FST), at American International University- Bangladesh (AIUB). His research interest includes the concept of Software Engineering Model & Process, Data Science, E-Learning, Machine Learning etc.

How to cite this paper: Shadman Latif, Faria Farzana Dola, MD. Mahir Afsar, Ishrat Jahan Esha, Dip Nandi, " Investigation of Machine Learning Algorithms for Network Intrusion Detection", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.14, No.2, pp. 1-22, 2022. DOI: 10.5815/ijieeb.2022.02.01