Annals of Emerging Technologies in Computing (AETiC)
Vol. 6, No. 3, 2022

Research Article

Comparative Analysis of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms

Johan Note and Maaruf Ali*

Epoka University, Albania
maaruf@ieee.org; jnote@epoka.edu.al
*Correspondence: maaruf@ieee.org

Received: 18th May 2022; Accepted: 14th June 2022; Published: 1st July 2022

Abstract: Attacks against computer networks, "cyber-attacks", are now common place affecting almost every Internet connected device on a daily basis. Organisations are now using machine learning and deep learning to thwart these types of attacks for their effectiveness without the need for human intervention. Machine learning offers the biggest advantage in their ability to detect, curtail, prevent, recover and even deal with untrained types of attacks without being explicitly programmed. This research will show the many different types of algorithms that are employed to fight against the different types of cyber-attacks, which are also explained. The classification algorithms, their implementation, accuracy and testing time are presented. The algorithms employed for this experiment were the Gaussian Naïve-Bayes algorithm, Logistic Regression Algorithm, SVM (Support Vector Machine) Algorithm, Stochastic Gradient Descent Algorithm, Decision Tree Algorithm, Random Forest Algorithm, Gradient Boosting Algorithm, K-Nearest Neighbour Algorithm, ANN (Artificial Neural Network) (here we also employed the Multilevel Perceptron Algorithm), Convolutional Neural Network (CNN) Algorithm and the Recurrent Neural Network (RNN) Algorithm. The study concluded that amongst the various machine learning algorithms, the Logistic Regression and Decision tree classifiers all took a very short time to be implemented giving an accuracy of over 90% for malware detection inside various test datasets. The Gaussian Naïve-Bayes classifier, though fast to implement, only gave an accuracy between 51-88%. The Multilevel Perceptron, non-linear SVM and Gradient Boosting algorithms all took a very long time to be implemented. The algorithm that performed with the greatest accuracy was the Random Forest Classification algorithm.

Keywords: ***cyber-attack; cyber defence; deep learning; intrusion detection system; machine learning***

1. Introduction

The current major problem that computer networks face are the many different types of cyber-attacks. This is the major problem that needs to be challenged, thwarted, mitigated, prevented and wherever possible full recovery made. The types of attacks range from those that cause no damage, such as adware (advertisement malware) to attacks that can steal your data (such as a phishing attack) to those that completely destroys data. Damaging and harmful attacks include those that can encrypt your computer in exchange for a ransom such as ransomware or attacks on the operating systems that can make it unusable, such as denial-of-service attacks. As the Internet-of-Things (IoT) devices are increasing in numbers so are the cyber-attacks increasing in scale and sophistication. Currently engineers and scientists are expending considerable research time in order to implement an intelligent system that will be employed for automated computer network intrusion detection. The aim of this project was to create an intelligent system that will detect the different types of anomalies in a network and show the performance of the methods investigated. Machine learning and deep learning classification algorithms were investigated due to their promising performance in unsupervised mode of operation. Both machine learning and deep learning are subsets of

Johan Note and Maaruf Ali, "Intrusion Detection System Using Machine Learning and Deep Learning Algorithms", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 19-36, Vol. 6, No. 3, 1st July 2022, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2022.03.003, Available: http://aetic.theiaer.org/archive/v6/v6n3/p3.html.

artificial intelligence. They are able to make the program learn without programming it in an explicit way. They are very useful in today's technology and they can be employed in every field of today's society. In our paper we will use supervised learning algorithms that are employed for classification. As libraries of machine learning and deep learning we will use scikit-learn and Keras respectively. Algorithms employed with scikit-learn are Gaussian-Naïve Bayes, Decision Tree Algorithm, Random Forest Algorithm, K-Nearest Neighbor Algorithm, Gradient Boosting, Multilevel Perception, Supporting Vector Machine both linear and non-linear etc. while with Keras we have used Artificial Neural Networks, Convolutional Neural Networks and Recurrent Neural Networks. Unlike machine learning algorithms in deep learning algorithms we use many layers one after another. Every layer has a specific activation function and can be employed to the specific layer. The latest techniques are CNN (Convolutional Neural Networks) and Recurrent Neural Networks (RNNs). CNN is usually employed for computer vision while the recurrent neural network is used usually for Natural Language Processing. CNN and RNN are the current state of the art.

Four different datasets were used to test the accuracies of the classification algorithms. The paper is structured into six sections: Introduction, Literature Review, Materials and Methods, Results and Discussion, Conclusions and Future Work. The introduction describes the project and its overall aim. The literature review is divided into three parts. The first part deals with the types of cyber-attacks and their detrimental effects. The second part presents the types of defences against cyber-attacks and who are the actual cyber criminals. The third part discusses the machine learning algorithms that are utilised in cyber security, forming the foundation of this research project. In the Materials and Method section, the datasets used for the project and the rationale is given. In the Result and Discussion part, the results of the training time, testing time, training accuracy and testing accuracy of the machine learning and deep learning implemented algorithms are presented in various tables. Finally, the Conclusions part gives the findings from this project and proposals for future work to be undertaken.

2. Literature Review

2.1. Cyber Attacks

A cyber-attack is a method to compromise the data in a computer of the victim by exposing, stealing, gaining, disabling or altering it without having legitimate access to it. A cyber-attack may also be considered an attack inside a system of a computer that is used to compromise the confidentiality, integrity and availability of the said data inside that computer. This is also known as the CIA triad. Confidentiality is the prevention of unauthorised users, access to the data. An example of the loss of the confidentiality is when someone who is not authorised, has access to, for example, the person's credit card number or password. Integrity is the security that a user that is not authorised, cannot modify the data. Availability is maintaining accessibility to the data every time it is necessary by the authorised user. An example of the loss of availability is when all the files in the PC are erased. There are many types of cyber- attacks, which are given in reference [1].

Sometimes unaware employees that work for an organization, may inadvertently give access to institution information to attackers. An example of misuse of resources attack, is a type of attack called, "The-Man-in-the-Middle" attack (MitM). This kind of attack takes place when a hacker puts himself between a reliable connection of a server and a client [2]. In this kind of attack, the cybercriminal modifies the communication being exchanged between the server and the client. Both the server and the client are completely unaware that a third person is between them and that he is intercepting and reading every message being transmitted. In this type of attack a cybercriminal puts himself between a session in the middle of the client (in this case the client is the victim) and the server. After that, the cybercriminal changes the client's IP (Internet Protocol) address and puts an IP address that he chooses himself. After that, he resumes the session with the server and the server unaware of everything sees the IP of the cybercriminal as a client that is to be trusted. Subsequently the cyber criminal's computer disconnects the client's computer and spoofs its number of sequence and the data inside the computer [3]. This type of MitM attack is known as session hijacking [4]. Figure 1, below, shows the categories of the different types of cyber-attacks.

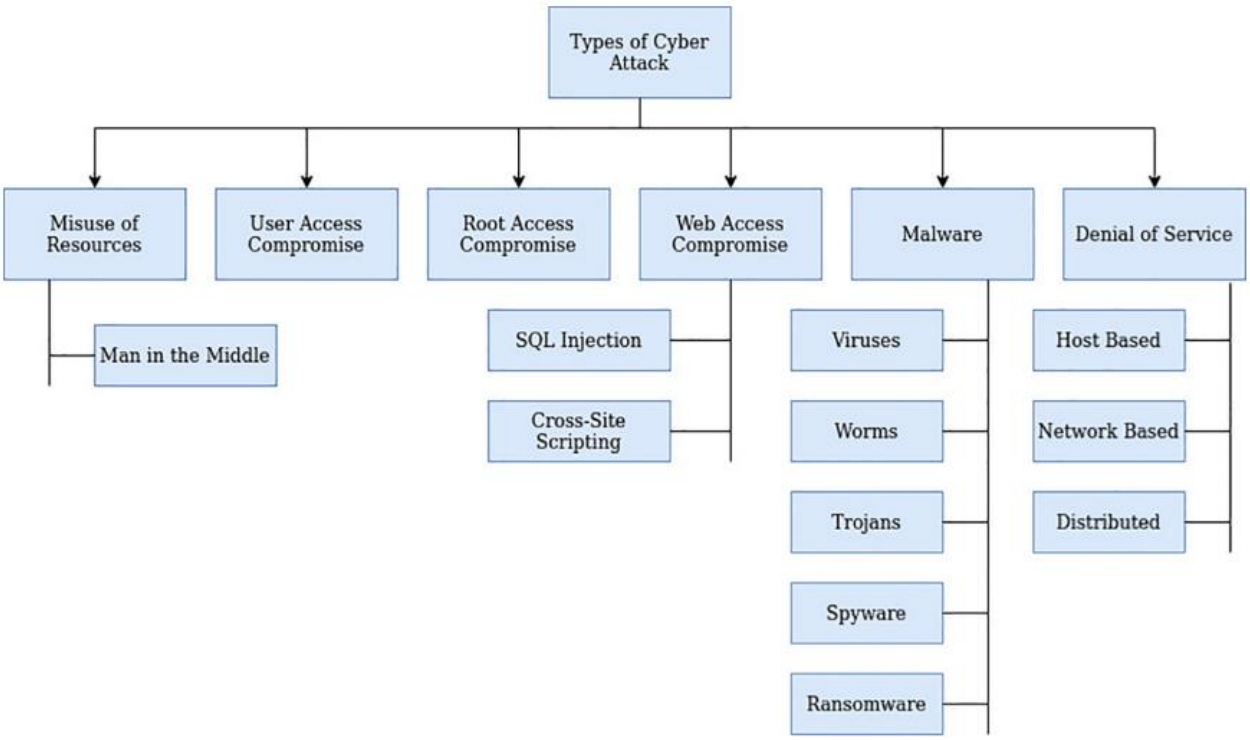

Figure 1. Types of Cyber-attacks [4].

A kind of attack that is usually used nowadays is compromising the user personal data, such as the credit card numbers, passwords etc. There are many methods used to compromise the user's personal data such as through social engineering, sniffing, brute force attack, dictionary attack, phishing, spearfishing etc. [2]. Sniffing is a method that a cybercriminal uses to capture data while it is transmitted from one PC to another through a wireless connection. There are many free software programs that help the cyber criminals to perform sniffing such as Wireshark. This kind of attack is usually used in places with public wireless access, such as in cafeterias. In this kind of attack, the cybercriminal is able to watch what is placed inside the website and what requests are being returned. Social engineering is a type of attack when a victim is psychologically manipulated to give data access to a cybercriminal. There are many ways how this can be done. For example, in class when the teacher asked the students how they had put the Facebook password, they started to show it one by one [3]. A brute force attack takes place when a cybercriminal tries to estimate the password of an account by attempting every feasible character combination until he has found it. A dictionary attack is a more advanced attack than the brute force attack. In this case like in the brute force attack, the cybercriminal tries to learn the password by attempting every feasible character combination, but in this case, he has a list of the most commonly used passwords such as '1234'. Phishing is a kind of attack when a victim is tricked into giving personal data access to the cybercriminal. Phishing is also known as stealing the personal information such as passwords and credit card numbers by a cybercriminal [3]. This type of attack has a great similarity to user access compromise, however, unlike in the user access compromise, in this attack the cybercriminal is able to obtain access to the account of the administrator and not in the individual host. As we know the administrator account has specific privileges in comparison to many others in the network system [4]. This type of attack is accomplished by utilizing vulnerabilities that are on different websites.

The most common methods to perform web compromising attacks are by SQL (Structured Query Language) injection and by XSS (cross site-scripting) [4]. SQL injection is that type of injection that permits the cyber criminals to compromise the database by spoofing of their identities, interfering with the data, creating declining problems such as damaging the information or rendering it unusable [5].

Malware (short for malicious software), is a software program that can make the computer malfunction. Hackers have used malwares for many years in order to accomplish their objectives for example: to turn off or to destroy a cyber-system, to compromise a system or network, to steal large-scale information and to inject harmful scripts [6]. According to the objectives and their propagation frequency they have, the malwares can be classified into several groups. The most common of these group are: worm,

spyware, virus, Trojan, ransomware [7]. Just as a biological virus in the human body, a computer virus is able to clone itself. It may be attached to a software program and after cloning itself, it is able to infect the other files on the computer. There are many types of viruses, examples include such as the Elk Cloner virus or the Melissa Virus [8]. Unlike the virus, a worm does not need a software program to clone itself. It can clone itself by propagating through the network. One way how the one worm can clone itself is by email attachment. Worms do not infect files on the computer. Worms are able to perform a Denial-of-Service (DoS) attack by cloning themselves through to every contact of the email of the victim and by using every network resource that is available [4]. In its purpose a Trojan is far different from viruses and worms. Cyber criminals who perform Trojan attacks usually use tricks of Social Engineering in order to fool the victim to install a Trojan in their computer. A Trojan neither infects the files on a computer nor is able to clone itself. Its only function is to establish a backdoor for cyber criminals in order to launch a malware when they have found entry [1]. A spyware is a type of malware that is employed to spy on the victim's activity rather than launch an attack. This kind of malware is used to take the user's data such as passwords, credit card numbers, login credentials, without awareness by the victim [9].

A Distributed-Denial-of-Service Attack (DDoS) is usually accomplished by a computer system or by a network in order to turn off the network of a victim completely [10].

Ransomware is a type of malware that blocks a group of software programs in a system in order to get a ransom (usually money). Usually, these kinds of attacks are accomplished with the help of a Trojan. An example of a ransomware is "Wannacry" [11].

In this kind of the cyber-attack, its principal aim is to destroy the normal operating state of a network or of a system. A Denial-of-service attack is divided into three main categories, which are: Host Based Attack, Network Based Attack and Distributed-Denial-of-Service (DDoS) attack.

Host Based Attack: in this type of attack, the inside of a computer system is infected with worms and malwares that are used to execute their operation or their payload in order to flood the whole network system with a hosts request's number that reaches infinity [12].

Network Based Attack: unlike the previous type of attack that needs a computer system, in this case, the cyber criminals target a whole network in order to run its payload and as a result of this, stop the normal operation inside the network [12].

This present research differs from [13] in that it utilises four different datasets unlike [13] which only used two, the kdd_test and nsl_kdd. The algorithms in the presented work also show a greater accuracy than [13], from 51% to 99% while the algorithms in [13] had a maximum ceiling of 91%. More algorithms have also been covered here.

2.2. Cyber-defences

There are several protective mechanisms in order to protect the system fully or partially against these types of attack. These protective systems are also known as "Intrusion Detection System" (or IDS). Intrusion Detection System contains an intrusion detection mechanism and an intrusion prevention mechanism. An IDS is created by a mixture of both hardware and software in order to monitor and manage network activities inside the network. According to the aim and the mechanism of the detection, the IDS is classified into two categories. The IDS has two methods for classification, which are: detection-based method and data source-based method. The detection-based method has two subgroups, which are: anomaly-based detection and misuse-based detection. The data source-based method has two subgroups, which are: network base method and source base method [14].

Detection based method: the signature-based detection is also known as misuse detection. The intention behind this kind of method is recognising the attack behaviours like signatures and storing them in a database. The misuse method is very fast and has a very low-level of false alarm rate. On the other hand, this method has a high-level of false alarm rate for unspecified attacks or no attacks at all [8].

Data source-based method: the host-based method is able to easily detect intrusion from a specific computer. This technique has the ability to detect the network object behaviour like files, ports and programs with a great precision. On the other hand, the host-based method is dependent on the host resources (computers in this case) and it cannot detect anomalies or attacks on the network. Unlike the host-based method, the network-based methods are independent of the host resources (computers). Network-

based methods are usually used on switches and routers. This system is operating system independent and can identify particular types based on the network protocol. One limitation is that it is used only to monitor information passing through a particular network [4]. An anomaly-based IDS is necessary in order to prevent misuse of resources attack on the network. This type of system is able to check the flows on the network and will raise an alarm if someone is in the process of a hijacking attack of any of the network sessions. Even though this type of system has a good performance for attack detection that are known to the network, it will still cause a high-level of false alarm rate for zero-day attacks. In order to have a network secure from zero-day attacks, it is suggested using preventive methods such as a Virtual Private Network (VPN) inside a network of a company in order to access the enterprise resources [9]. A case of user access compromise as well as root access compromise is the phishing attack. This means that by protecting and by preventing against a phishing attack we are able to secure both the root access and user access on a particular system. One method to secure it is by using an email-based method as a protecting mechanism in case of a phishing attack[1]. SQL injection attack and XSS attack are used by cyber criminals in order to initiate a web access compromise attack against a web address when they see a vulnerability. Two detection methods are used to protect the system in case of a web compromise attack which are signature-based detection method and anomaly detection method. In addition other measures that are able to prevent recent knowledge in vulnerabilities inside the website include keeping actual patches used for applications up to date. It helps to have a practice for secure coding in order to block vulnerabilities inside the database [2]. Currently malware is a pandemic that has plagued all the cyber world. It is the primary choice of method that is used by cyber criminals to attack computer systems. In order to protect against an attack from malware, techniques of malware detection are the first line in the defensive system. According to the way how malware is going to be handled for detection, the mechanism for detection is divided into three groups.

Signature based: this method that is used for malware detection is very frequent. Companies that produce anti-malware examine the malware and later generate signatures (signatures are a sequence of bytes). The signatures are employed in order to issue security to their clients according to a pattern-matching algorithm. Nevertheless, the primary restriction of this method is that cyber criminals may change a chunk of code and of the preceding malware in order to avoid signature-based detection systems. Also, this method is unsuitable on zero-day attacks [11].

Behaviour based: even though the idea of behaviour-based malware detection is much related to the idea of the signature-based detection, this technique has a different approach of feature extraction. In this technique, the mechanism of the detection counts the actions of the malware, not what it says. The behaviour-based malware detection technique is appropriate for detection of malwares that may obfuscate and malwares which are mutant. Rather than issuing different signatures at different byte code patterns, malwares with related habits are categorised into a single signature. This remarkably minimizes the false alarm rate of behaviour-based methods. The behaviour-based detection approach has three different items. The first item which is the data collector, is used to collect the data regarding the information on the executable element. The other item operates like an intermediate medium that is used to transform the collected data to an intermediate medium. Finally, the representations are matched to the database of the behaviour signature in order to issue the output [4].

Heuristic based: even though the behaviour-based detection approach is far more powerful than the signature-based approach, the cyber criminals are still able to avoid this method by taking strong counter-measures. In order to overwhelm this obstacle, today's researchers use a Data Mining-based approach and Machine Learning Learning-based approach, called the heuristic method [2]. Protection against the DoS (denial of service) attack is divided into two groups, which are known as attack prevention and attack detection [5].

Attack prevention: this method is mainly used in networking routers in order to identify malicious traffic according to signatures. In case of a DoS attack, the attack prevention method is also as the first line of defence. Some of the methods used for filtering packets are: ingress/egress filtering. This filtering of packets controls the network traffic on the condition that ingress traffic (traffic that comes inside the local

[1] "Support Vector Machine Algorithm", https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

network) or egress traffic (traffic that goes outside the local network) is equivalent to that which is expected from the source IP.

Router-based packet filtering: this type of filtering is accomplished according to routing information for the arriving packets with respect to their source and destination IP addresses [12].

Packet-filtering based on Hop Count: in the network traffic the difference between the initial value and the Time-to-Live (TTL) value in a packet is called the hop count. In this method, a network router composes a database table for each user and its matching hop count to any specific destination. Therefore, in case that the router discovers any kind of anomaly in the anticipated hop counts, it will drop the packet and will raise an alarm in order to protect the network against a potential threat or attack [12].

Attack detection: this type of detection mechanism is used against the DoS attack, being divided into two groups which are: Signature-based detection and Anomaly-based detection.

Signature-based detection: in this technique, the malicious traffic is identified according to the signatures of the data of attack traffic [4].

Anomaly-based DoS detection: this technique is generally used for DoS detection, because the attack patterns are more complex than before. Machine Learning techniques are generally used for this technique. This method is composed of two primary factors. Firstly, network features like Time-to-Live (TTL), length of IP packet etc., are extracted from the network traffic data by employing Data Mining (DM) techniques and after that a model for detection is constructed on that feature depiction. Secondly, the arriving traffic is passed with this model and according to the value of a preselected threshold, it determines if the traffic is malicious or not [12].

2.3. Machine Learning

Machine Learning is a blanket term that is employed for computational approaches which attempt to imitate learning activities of humans via computers to automatically find and obtain knowledge. It is a wide-scale research area of study and includes many fields such as: psychology, neuroscience, computer science and statistics [1]. Today, the learning algorithms have improved remarkably. This is a result of current developments in the speed of processors and big data. According to the learning techniques, algorithms used in Machine Learning are grouped into three different categories which are: supervised learning, unsupervised learning and reinforcement learning. Figure 2, show the family of ML algorithms.

In supervised learning algorithms, we train the models to such a degree that the given true output labels are mapped in order to learn the relation with their matching feature value. Some examples of supervised learning algorithms are: Logistic Regression, Support Vector Machine (SVM), Random Forest[2], Decision Tree, K-Nearest Neighbour but also the neural networks such as ANN (here it is also included the Multilevel Perceptron), Convolutional Neural Network[3] (CNN) and Recurrent Neural Network[4] (RNN) [5]. On the other side, unsupervised learning algorithms learn the data based on the entire training dataset, even not knowing the output for each input. Unsupervised Learning algorithms do not have any labels in their training data. K-means clustering[5] is the one best examples of the unsupervised learning algorithm category. A reinforcement learning (RL) algorithm focuses on learning from the environment where it places an agent. Here the agent is able to learn from actions that takes place inside the environment and based on these actions, it will make an error or an achievement. Reinforcement learning is a combination of both supervised and unsupervised learning algorithms [15].

[2] Tony Yiu, "Understanding Random Forest", https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[3] Sumit Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", 15 December, 2018, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[4] Jason Roell, "Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data", 26 June, 2017, https://towardsdatascience.com/understanding-recurrent-neural-networks-the-prefered-neural-network-for-time-series-data-7d856c21b759.

[5] Victor Lavrenko, "K-means clustering: how it works", 20 January, 2014, https://www.youtube.com/watch?v=_aWzGGNrcic.

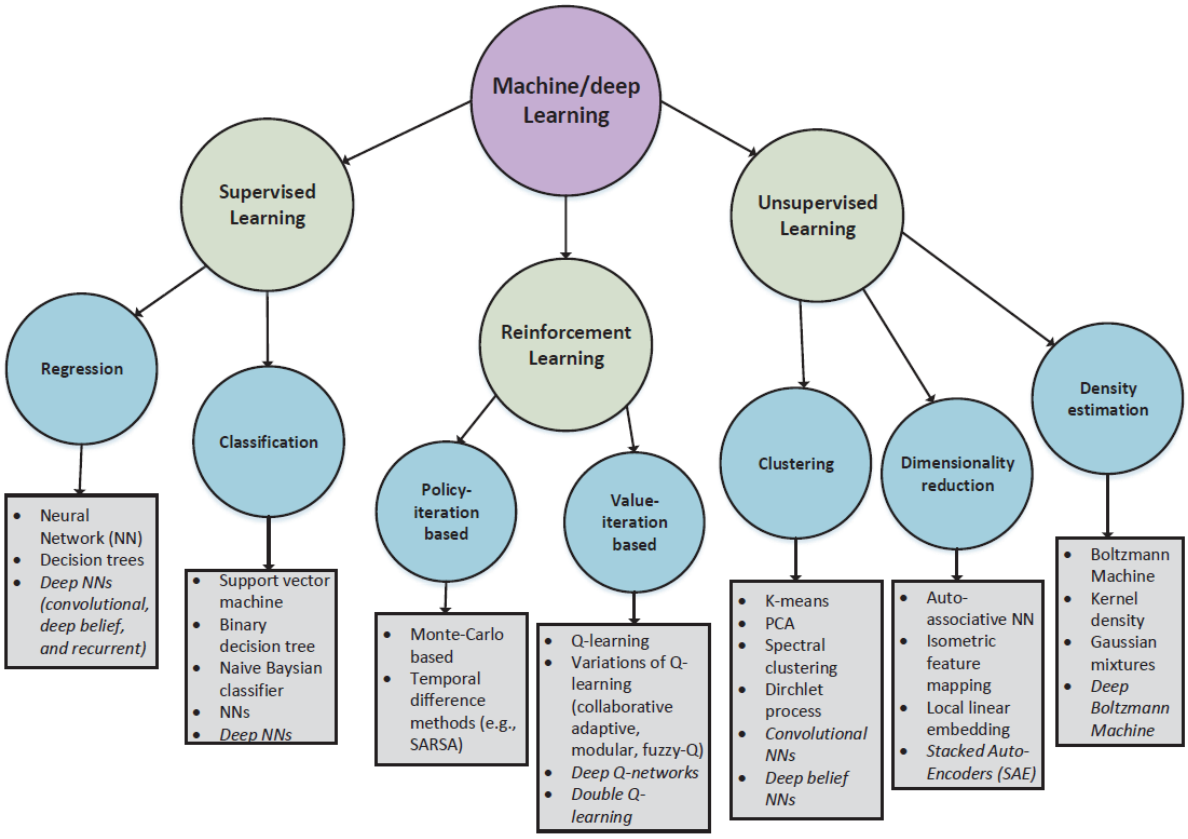

Figure 2. The Family of Machine Learning Algorithms [16].

The Decision Tree algorithm is a rule-based model that is used for classification. It has a tree structure in which every vertex of the tree represents a feature while every branch decides the feature value. The vertex which is on the top of the tree is known as the root. It holds the majority of the information gain (entropy differences) among all the features and is employed to properly split the data that is going to be trained. The vertices at the bottom of the algorithm are known as the leaves. A leaf is the representation of a class. In the course of the classification, the decision tree crosses to a top-down approach in order to satisfy the instance that is necessary for classification [4]. The information gain equation that is employed in a decision tree to particularly split samples in a tree-structured method is as shown in (1) [7], below:

$$Gain(P, Q) = Entropy(P) - \sum_{v \in D_Q} \frac{P_v}{P} Entropy(P_v) \quad (1)$$

Where $Gain(P, Q)$ is the entropy reduction that is used to sort P on feature Q. Features which have an information gain value that is increasing are selected as nodes in a top-down method. The decision tree algorithm's primary advantage is that it is simple to implement and provides a high level of classification accuracy. On the other hand, the decision tree classifier's primary disadvantage is its computational complexity [7].

The Random Forest algorithm is composed of many decision trees which work as an ensemble. Every tree which stays inside the Random Forest is able to make a class of prediction and after all the Decision Tree's output are gathered, they are analysed to determine the most voted class, which will then be the model's prediction. This type of algorithm may be employed either for classification or for regression. This method establishes additional randomness when trees start to grow. Rather than looking for the best attribute at the moment of dividing a node, it looks for the best attribute from a random subset of attributes. Because of this the outcome of the random forest is a more significant diversity of trees, which deals with a higher-level bias for a lower-level variance, usually giving a far better model [14].

The Logistic Regression algorithm is an algorithm that is frequently employed in order to calculate the probability that a given sample belongs to a particular class (for example if this file is a virus or not). In case that the calculated probability is greater than 50% then the training model makes a prediction that the given sample is part of that class (it is named the positive class sample, being labelled with the value 1). In the

other case, when the calculated probability is less than 50%, the training model makes a prediction that the given sample is not part of that class (it is named the negative class and the sample takes the value 0). These calculations make the Logistic Regression a binary classifier. A Logistic Regression Algorithm is able to calculate a weighted sum of a group of input attributes (in addition of a bias), but in this case the output is the logistic combination of the result rather than the result itself. The Logistic function is a sigmoid function and is represented as σ(.), whose output is a number from 0 to 1 [14]. The Logistic function is shown in (2) [14], below:

$$\sigma(t) = \frac{1}{1+e^{-t}} \quad or\ it\ can\ be\ written\ as\ \ Y = \frac{1}{1+e^{-x}} \tag{2}$$

From which x (or t) is the input of the function, Y (or $\sigma(t)$) is the output of the function.

Support Vector Machine (SVM) is one of the most favoured algorithms in the field of cyber security. The especial thing in this algorithm is that it uses a hyperplane to divide the different classes. The hyperplane that this algorithm uses, is to such a degree the interval between that and its nearest data point which is to be maximised. This algorithm can be employed both in the 2D plane and 3D plane. The aim of the SVM is to classify the data precisely. Some of the SVM advantages are that it is easy to be implemented, with a very high rate of accuracy and with the ability to generate a hyperplane with time complexity. A disadvantage of using this method is that it is difficult to choose an optimal kernel size. This algorithm is employed in many different fields such as in medicine, security applications, pattern recognitions etc., [17].

Another supervised algorithm which is a probabilistic-based algorithm is the Naïve-Bayes[6] classifier. This algorithm issues the probability of a class when all the attributes are given as inputs. This algorithm is modelled according to the Bayes rule. The Naïve-Bayes algorithm is also known as the generative model. In order to compute the probability of a class which is $p(b/a)$, the Naïve-Bayes classifier computes the conditional probability of all attributes that are given in a class which is $p(a/b)$ with the initial probability of all classes which is $p(b)$. The general formula for Naïve-Bayes algorithm is (3) [17]:

$$p\left(\frac{b}{a}\right) = \frac{p(a,b)}{p(b)} = \frac{p\left(\frac{a}{b}\right)p(a)}{p(b)} \tag{3}$$

In which "**a**" is the input vector while "**b**" is the class vector. The Naïve-Bayes classifier's primary advantage is that it is strong with noisy training data. Because this type of classifier is based on the probabilistic values of all attributes, low-level training samples with the Naïve-Bayes classifier do not deteriorate its accomplishment. The primary disadvantage of the Naïve-Bayes classifier is that all the attributes are considered being independent, even though in practice this barely takes place [18].

The K-Nearest Neighbour algorithm, also known by its acronym KNN, is a well-known algorithm for its simplicity when it is executed in a program [19]. This algorithm is very useful when it is employed either in classification problems or in regression problems. It is one of the most well-known supervised algorithms. It is employed in many fields of technology today. The k-value inside the K-Nearest Neighbour is dependent on the dataset size and the type of the problem that is used for classification or for regression [19]. An equation that is employed for variables that are continuous is the equation of the Euclidean distance. In order to discover the nearest features from a data that is going to be tested x to a data that is going to be trained k according to the Euclidean distance in order to compute the length of the distance. According to the Euclidean distance when two features are given in dimensional space of k, $x = [x_1, x_2 \dots \dots, x_k]$ and $y = [y_1, y_2, \dots \dots, y_k]$ then, the equation of the Euclidian distance according to these two features will be calculated as shown below by (4) [19]:

$$D(x,y) = \sqrt{\sum_{i=1}^{k}(y_i - x_i)^2} \tag{4}$$

After the accumulation of all the KNN data is finished, the KNN majority is going to be regarded as a class for the data that is going to be tested [19].

The ANNs are made up of nodes, which are influenced by the brain neurons. An ANN should have at least three layers consisting of the input layer, a hidden layer and an output layer [4]. It may have more than a hidden layer inside, depending on its network design. The output of the input layer passes through to the hidden layer and the output of the hidden layer passes to the output of the next layer and so on. In

[6] "How is Naive Bayes a linear Classifier?", https://stats.stackexchange.com/questions/142215/how-is-naive-bayes-a-linear-classifier.

the course of the learning process inside the ANNs, inputs (x_1, x_2, x_3 … $x\_(n\text{-}2)$, $x\_(n\text{-}1)$, x_n) are offered with output an output label of value y, in which the information taken by the input is weighted by a weight vector (w_1, w_2 … w_(*n*-2), w_(*n*-1), w_*n*) [4]. All during the time of the learning process, the weights are modified to such a degree that they reduce the learning error. Equation (5) [14] shows how the error is calculated:

$$E = \sum_{i=1}^{n} |d_i - y_i| \tag{5}$$

In which d_i is the output that is required, y_i is the output that is known and E which in this case is the difference between the output that is required and the output that we have, i.e. the error. The modification is made possible with the help of a gradient algorithm known back-propagation, in which the learning process rehearses backward and forward till the model gets an error value less than its threshold value [1]. The weighted vector is modified based on (6) [7]:

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j} \tag{6}$$

in which "*i*" is the input node while *j* is the hidden node.

Another class of deep learning algorithms is the CNN. It is generally employed in order to manage large training datasets by employing hierarchical-based attribute abstraction and representation [20]. The implementation of normal Machine Learning algorithms deteriorates in the case where the dataset is very large or because of the dimensionality of the data. In order to manage this problem, Deep Learning is being utilised with the assistance of graphic processing units (GPUs) in order to process the large data. From all the deep learning algorithms, the CNN is being employed considerably in applications of cyber security [20]. The two primary layers of the CNN are the convolutional layer and the pooling layer. The convolutional layer is used to convolve the input data with the assistance of multiple kernels which have the same size. The operation of the convolution recovers the attributes from the input data by issuing a high value for a random position if the desired attribute is available and vice versa [3]. Equation (7) [3] is employed to find the expected value:

$$h = \sum_{k=1}^{m} \sum_{l=1}^{m} w_{k,l} x_{i+k-1,j+l-1} \tag{7}$$

h is the output of the convolution, w is the convolution kernel and x is the input. The next layer which is the pooling layer is utilised in order to down-sample the size of the feature by using two types of pooling technique which are max-pooling and average pooling. The max-pooling selects the biggest value in the features, while average pooling is employed to take the average values. CNN employs an activation layer which is known as a Rectified Linear Unit (ReLu) that contain perceptrons with an activation function which is shown in (8) [3]:

$$f(x) = \max(0, a) \tag{8}$$

One disadvantage of the CNN is the cost. Another disadvantage of this algorithm is the time it takes to be implemented because of the number of layers.

Different applications in which the present state output depends on the state of various previous states, traditional machine learning algorithms do not issue a good performance. This takes place because in these types of algorithms, they do not have inter-dependency between the output and the input. RNN which is another Deep Learning algorithm specifically manages these kinds of sequential data and appears a better accomplishment than all the other algorithms. Even the RNN is composed of at least three layers which are the input layer, the hidden layer and the output layer. Data flows in RNN take place only one way: from the input layer to the hidden layer. A combination of this one-way data flow takes place from a previous sequential disguised layer to the present time hidden layers. All of the data all over the RNN is stored in the hidden layers. The RNN calculates the vector sequence of the hidden layer which is $h = (h_1, h_2, h_3 \ldots h_{(N-1)}, h_N)$ *in* order to compute the vector of the output layer which in this case is $y = (y_1, y_2, y_3 \ldots y_{T-1}, y_T)$ [12] from $t = 1$ to $t = T$ iteration of the equations (9) [12] and (10) [12]:

$$h_T = H(W_{xh} x_t + W_{hh}\, h_{t-1} + b_h) \tag{9}$$

$$y_t = W_{hy} h_t + b_y \tag{10}$$

in which W is the representation of the Weight matrix, b is the representation of a bias vector and H is the representation of hidden layer [12].

3. Materials and Methods

3.1. Datasets

There are four datasets used for this experiment, which are: kdd cup dataset, nsl-kdd dataset, Kyoto dataset and UNSW-NB15 dataset. The reasons why these datasets are used is because these datasets are very well structured and are very useful in these kinds of experiments especially when we discuss about machine learning algorithms. Another reason for using these datasets is that these datasets are free to use. It is easy to use these datasets. For data science these kinds of datasets have a very good quality.

The KDD cup dataset is composed of around 4.9 million vectors which are single-connection and every vector is composed of 41 attributes and can be labelled like normal or like an attack. Inside a KDD cup dataset there are four categories of attacks [17].

Denial of Service: in this type of attack, the device memory is busy and full and in such a way that it cannot respond when it takes the request [18]. The best way to defend from this kind of attack is by turning the device off.

User to Root Attack: in this type of attack, the cybercriminal which has a specific access to a device wants to exploit the system vulnerabilities in order to have access to the router. This can be done by using many different methods such as phishing attacks, sniffing also known as packet controlling or by using social engineering.

Remote to local attack: this type of attack takes place the moment when the cybercriminal, which does not have access to the device, is able to deliver packets from a computer device to a network system and is able to exploit the system vulnerabilities in order to have access to the computer device.

Probe attack: this type of attack is an effort to obtain data over a computer network system for the obvious aim to avoid the system's controls of security.

The kdd dataset attributes are divided into three categories which are: basic attributes, traffic attributes and content attributes. The basic attributes category encapsulates every feature which may be removed from an IP/TCP connection. The majority of all these attributes lead to a complete detection delay. The traffic attributes category contains attributes which are calculated regarding a window interval and is subdivided into two subcategories which are ''same host attributes'' and ''same service attributes''. Same host attributes are used to inspect the connections in the previous two seconds whose destination host is the same as current connection. This attribute is also able to compute statistics which are affiliated to the behaviour of the protocol. Same service attributes are used to inspect the connection in the previously two seconds whose service is the same as the current connection. Both of these subcategories of traffic attributes are also named as time-based. Nevertheless, there exist various slow-moving probe attacks which are able to scan the ports (or hosts) by employing time intervals that are larger than two seconds, for instance there may be one for each minute. Therefore, these kinds of attack do not create patterns of intrusion for a time window which is two seconds. So, in order to resolve this problem, the attributes of same host and same service are recomputed and according to the connection window which is composed of 100 connections. These kinds of attributes are also known as connection-based traffic attributes. Although we have different types of attacks such as User to Root Attack and Remote to Local Attack which have no frequent sequential patterns of intrusion [15]. These types of attacks are inserted in the packet's data portion and usually have only a single connection. In order to detect these types of attacks, it is necessary to have several attributes that can be inspected for the behaviour that is suspicious inside the data portion. These attributes are also known as content attributes.

The NSL-KDD dataset is similar to the KDD dataset. This dataset is composed of four different subcategories which are KDDTest+, KDDTest-21, KDDTrain+, KDDTrain-21. KDDTest+ and KDDTrain+ are full datasets with all the components while KDDTest-21 and KDDTtrain-21 compose only 20% of the respective dataset. The NSL-KDD datasets are composed of 43 attributes each. In every dataset the first 41 features are referred to as the traffic input while the other two are referred to as Label (if this traffic input is an attack or not) and Score (the ferocity of the attack traffic input). As was explained before, there are four categories in this dataset which are completely similar to that of the KDD dataset which are Denial of Service attack, Remote to Local attack, User to Root attack and Probe attack. The attributes in the record of traffic issue data in order to confront the traffic input with the help of an IDS and are divided into four groups which are Intrinsic attributes, Content attributes, Host-based attributes and Time-based attributes. Intrinsic

attributes are used to carry the basic information that is available for the packet. In the dataset the intrinsic category carries attributes from 1 to 9[7]. Content attributes carry data concerning original packets. These attributes are forwarded in many chunks rather than only one chunk. By the help of this data, the network system has the ability for accessing the payload. In the dataset, the content category carries attributes from 10 to 22. Time-based attributes carry the traffic input analysis over a window for the last two seconds and is composed of data alike the number of connections that it tried to create in the same host. In this category the attributes are mostly rates and counts. Host-based attributes are like the time-based attributes, but they do not analyse the traffic over a window for the last two seconds. Instead of that they analyse the traffic over sequences of connections made. The host-based attributes are created for accessing attacks whose span is longer than two seconds. In the dataset, the host-based category carries attributes from 32 to 41[8].

The Kyoto dataset is a dataset which was created at Kyoto University. It was created by using real traffic data. This type of dataset is composed of 24 statistical attributes, from which 14 attributes were extracted. From these 14 attributes, ten others are appended. This dataset shows a great level of accuracy. This type of dataset was made available by using honeypots, web crawler, darknet sensors and email servers [21]. In the output of Kyoto dataset it controls whether there was an attack or not. There are three indicators in this type of dataset. The first indication is when it is '1' so there is no attack. The second indication is when there is '-1' that means there is a known attack and the third indication is when there is '-2' that means that there is an unknown attack.

Another dataset that was used in this program is the UNSW-NB15 dataset. This dataset is composed of 42 attributes. From all these attributes them of them have a value which is considered categorical (means non-numerical) while the other 39 attributes contain a value which is numerical [11]. The reason why this type of dataset was chosen was because it is a good dataset for checking an intrusion detection system. In UNSW-N15 datasets there are 9 types of attacks which are: Fuzzers, Analysis, Backdoors, DoS (Denial of Service), DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

The NSL-KDD cup dataset and KDD-cup dataset are similar to each other in terms of components related to both the inputs and outputs but they do not have any similarity with Kyoto or UNSW-N15. The differences are still of use in helping to train our classification algorithms.

3.2. Machine Learning Algorithms

The algorithms employed for this experiment are the Gaussian Naïve-Bayes algorithm, Logistic Regression Algorithm, SVM (Support Vector Machine) Algorithm which are both linear and non-linear, Stochastic Gradient Descent Algorithm, Decision Tree Algorithm, Random Forest Algorithm, Gradient Boosting Algorithm, K-Nearest Neighbour Algorithm, ANN (here we also employ Multilevel Perceptron Algorithm which is a subset of ANN), CNN Algorithm and RNN Algorithm.

The Gaussian Naïve-Bayes algorithm as we have mentioned earlier is based on the Bayes theorem. This algorithm works in a number of steps. First of all it is necessary to compute the previous, i.e. the earlier likelihood for all the class labels that are given to us. After that we need to generate a table based on the prior data that we had and after that to see how frequent every phenomenon happens. In the next step we need to determine the likelihood probability for every one of the features in every one of the classes. After doing all of these we need to set all these values that we have computed inside the Bayes Formula in order to compute the posterior probability. After doing all these steps we look at the class which has the highest probability as that will be the output of the sample. This algorithm takes a short time to be implemented, however, it has a value of accuracy rate smaller than the other algorithms[9].

Logistic regression is another algorithm employed for classification. In this kind of experiment that was implemented using the Python programming language, we have a Multinomial Logistic Regression, because there are more than two outputs to come. This type of algorithm is an expansion of the Binary Logistic Regression. Similar to the Binary Logistic Regression, the Multinomial Logistic Regression employs

[7] Z._Ai, "Logistic Regression Explained", https://towardsdatascience.com/logistic-regression-explained-9ee73cede081.

[8] Ibid.

[9] Avinash Navlani, "Decision Tree Classification in Python Tutorial", https://www.datacamp.com/tutorial/decision-tree-classification-python.

the maximal likelihood evaluation in order to estimate the categorical shift probability. For Multinomial Logistic Regression it is not necessary to have a cautious examination for the sizes of models and consideration for obscure occasions. It is employed in many different fields such as in medicine, statistics, etc. [22]. Now we are going to see how this algorithm is trained. The aim of training is to place the vector denoted as $\boldsymbol{\theta}$. Hence the model calculates a high-level of probability for the samples that are positive and a low-level of probability for the samples that are negative. The cost function of a sample trained x is given by (11) [9]:

$$c(\theta) = \begin{cases} -\log(p) & if\ y = 1 \\ -\log(1-p) & if\ y = 0 \end{cases} \tag{11}$$

In which p is the probability found and y is the output. This kind of cost function is reasonable because -log(x) increases a lot as the input value x comes near to zero. Hence the cost is going to be very large in case that the model calculates a probability which is near to 0 for a positive sample. But it is also going to be very large in case that the model calculates a probability that is near to 1 for a negative sample. However, $-log(x)$ is near to 0 when x is near to 1. Hence the cost will be near to 0 in case that the calculated probability is near to 0 for a negative sample and will be near to 1 for a positive sample, that is what we exactly want [15]. The cost function of Logistic Regression is shown in (12) [6]:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(p^{(i)}\right) + \left(1-y^{(i)}\right)\left(\log\left(1-p^{(i)}\right)\right)\right] \tag{12}$$

In which m is the number of all the samples, while $\sum_{i=1}^{m}\left[y^{(i)}\log\left(p^{(i)}\right) + (1-y^{(i)})(\log(1-p^{(i)}))\right]$ is the sum of all the training sample costs. Unfortunately, there does not exist a type of closed-form equation which will be used to calculate the θ value which is used to minimize the cost function. However, this type of cost function has an advantage in that it is convex, hence every algorithm that is used for optimisation is promised to discover the global minimum. The equation for the cost function partial derivatives is given by (13) [9], below:

$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\sigma(\theta^T * x^{(i)}) - y^i)x_j^{(i)} \tag{13}$$

As we see from (13), for every sample it calculates the prediction error and after that it multiplies it with the j attribute value and after that it calculates the average of all the training samples. It shows a great level of accuracy for both the training and testing datasets, it also takes a very short time to be implemented. The reason why this algorithm was chosen was because it is very precise.

According to the cyber security viewpoint, tracks of anomalous behaviour can be detected in the case that network logs can be scanned. In order to divide the anomalous behaviour from the licit ones, a strategy based on classification is necessary to be installed inside a networking system. One of the most uncomplicated algorithms is also the Decision Tree algorithm. It shows a great level of accuracy compared to the Gaussian and other algorithms. The Decision Tree primary concept is to divide the dataset based on the data gain of the attributes. The library scikit-learn also known in short in Python as sklearn[10] employs an algorithm known as the "Classification and Regression Tree" algorithm for training of the Decision Trees (which are also known as "growing trees"). The concept behind this algorithm is very simple: the training algorithm firstly divides the set that is going to be trained into two subsets by employing a unique attribute k and a corresponding threshold for this attribute which is t_k. But now it raises the question on which way it selects the parameters of the attribute k and the threshold t_k. The algorithm looks for the set (k, t_k) which is able to create the most clarified subsets. The equation for the algorithm that attempts to reduce the cost function is given below, as (14) [1]:

$$J(k,\ t_k) = \frac{m_{left}}{m}G_{left} + \frac{m_{right}}{m}G_{right} \tag{14}$$

From which m_{left} is used to estimate the left-wing subset impurity, m_{right} is used to estimate the right-wing subset impurity, G_{left} represents the sample numbers of the left-wing subset, the G_{right} represents the sample number of the right-wing subset while m represents the number of all the samples. At the moment that this algorithm has divided the set that is going to be trained into two different subsets, it starts to divide the subsets by employing the identical logic, after that it starts to divide the subsets of the subsets in the

[10] "sklearn.linear_model.SGDClassifier", https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

same way and so on, in a recursive way. This algorithm halts the recursion the moment that it reaches the maximal depth, or in the case that it is not able to discover a division that is able to decrease the impurity [4].

A group of Decision Trees is called a Random Forest Classifier. It has a great level of accuracy as well as it takes a very short time usually a very few seconds in order to be implemented. In this case, there are 30 trees employed as inputs. It shows excellent performance compared against the other algorithms.

As mentioned previously, the SVM is able to divide the different classes of components by the help of hyper planes. SVM classification algorithms are divided into linear SVM classification algorithms and non-linear SVM classification algorithms [23]. Linear SVM classification algorithm classes are split between each other by the help of a straight line. This classification algorithm is usually used for 2-dimensional planes. In linear SVM classification, the straight line is used not only for splitting different classes, but on the other hand it is able to be placed some distance from the nearest training samples, based on probability distribution and clustering. The SVM classification algorithm fits the most extensive possible way which stays between these different classes that this algorithm has divided. This is also known as large margin classification. The decision boundary will not be modified by the addition of more training samples. This boundary is decided by the samples that are located at the borders. There samples that are located at the borders are also known as the supporting vectors from which this algorithm takes the name. Hard margin classification is the forcing of all samples to be off the way on its right side. By employing hard margin classification there are two primary concerns [11]. Firstly, it is only able to work in case that the data is separable linearly and in addition it is very sensitive in response to outliers. In order to evade these concerns it is better to employ a model which is more flexible. The aim of this model is to discover a fine stability between maintaining the way as large as it is the possibility and restricting the violations of the demarcating margin. This is also known as soft margin classification. In SVM classes of scikit-learn, we are able to manage the stability by employing the hypermeter *c*. The advantage of this method is that linear SVM in scikit-learn show a higher accuracy between 95% and 99%. However, this method takes a lot of time to be implemented. Even though the linear SVM classifier models have a high efficiency rate and superior performance in a lot of cases, a great number of datasets are not close to being linearly separable. A technique in order to handle with these types of datasets is by adding more features, like the polynomial feature.

The non-linear SVM classifier is usually employed in 3-D models. In this classifier, the hyper plane is applied differently from the linear 2-Dimensional plane. When we employ the SVM algorithms, it has a mathematical method known as the kernel trick. This method is able to make it feasible to obtain the same result like in case that you have added a lot of polynomial features, even in case that the polynomial may be of a very high degree, without being necessary to add all these polynomials [23].

Another model that can be employed for linear SVM is the Stochastic Gradient. This is an estimator that is used to perform models which are linear (in this case Linear SVM) with Stochastic Gradient Learning (SGD) learning. SGD learning means that the loss gradient is computed after every sample inside a given time and after that the model will be updated alongside the process with a schedule of decreasing strength. It is a very powerful classifier and in this algorithm we can determine the level of tolerance that this algorithm may have[11].

Gradient Boosting algorithm is a Boosting algorithm that is very popular. This type of algorithm operates by adding the predictors inside an ensemble by using a sequential way. Every predictor that is added inside the ensemble is used to correct the predecessor that it had inside the sequence. Although, rather than modifying the sample weights at each iteration, this technique attempts to fit the current predictor with the residual errors which are done by the preceding predictor [15]. This type of algorithm is beginning to be well-known as a result of its efficacy that it has for complex datasets. The Gradient Boosting algorithm can be employed in both regression and classifications problems.

K-Nearest Neighbour is a supervised learning algorithm. It is imported from the sklearn algorithm. It takes a lot of time to be implemented as an algorithm. The variable *n_neighbours* inside the code is the acronym of the number of neighbours. This is a variable employed for the prediction of the model. In this code the number of neighbours is five, because of the small number, it shows great accuracy. The distance

[11] Ibid.

used in this classification is the Minkowski distance. This type of distance can be computed only if distances are regarded as vectors inside a space and these vectors should have a specific length. The small *p* represents the parameter of power which in this case when we are employing Minkowski distance it should be 2. The K-Nearest Neighbour classifier has one advantage, that is, it is able to show a great level of accuracy. On the other hand this kind of classifier takes a lot of time to be implemented and it is not good when there is a very large dataset.

The ANN classifier as mentioned previously, is made up of nodes (also known as perceptrons), which have been inspired by the neurons of the brain. The training of this ANN in Python is made available by a library known as Keras. This is a library that is employed in order to train the neural network algorithms. From these algorithms the simplest one of them is the ANN [15]. In our experiment the ANN algorithm that has been trained has three layers: the first layer is the input layer; the middle layer is the hidden layer and the third layer is the final output layer. Inside each layer, we put the number of dimensions and the activation function of the layer [5]. The activation function of the input layer in this case is the ReLu. The activation function of the hidden layer in our experiment is a sigmoid function, while the activation function of the output layer is a Softmax function. The ReLu activation function is an activation function that makes all the negative functions 0 while retaining the positive values [20]. The sigmoid activation function is an activation function that makes all the values have values between 0 and 1 [20]. Because of this reason, this activation function is very useful for doing probability prediction. Equation (15) [8] is the formula for the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{15}$$

From which z is the input and $\sigma(z)$ is the output. The maximal value of the output $\sigma(z)$ is when the input value z is equal to infinity. The Softmax activation function is a function that is employed in order to normalise a value of an input inside a values vector that has a distribution of probability from which the whole probability is estimated to be up to 1 [8]. Even in this case the output values are between 0 and 1. This type of activation function is employed in order to calculate the losses which are expected at the moment when we train a given dataset. The equation of Softmax activation function is (16) [8]:

$$P\left(y = j \middle| \theta^{(i)}\right) = \frac{e^{\theta^{(i)}}}{\sum_{j=0}^{k} e^{\theta_k^{(i)}}} \tag{16}$$

From which (17) and (18) are derived [8]:

$$\theta = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_{k-1} x_{k-1} + w_k x_k \tag{17}$$

$$\sum_{i=0}^{k} w_i x_i = w^T x \tag{18}$$

We employ the softmax activation function in our Python program, because we have more than two different outputs.

The CNN is another form of deep learning algorithm. This algorithm is good for making image classification. In the program of this paper, the CNN classifier is composed of nine layers. Before starting the CNN classification, the dataset is reshaped to four dimensions (from two dimensions that it was). The input layer which is a 2-Dimensional Convolutional layer, the activation function is the same as the previous algorithm's ReLu. In our equation it convolves values which is inside 3×3 dimensions. The input shape is the same as the three reshaped dimensions that we have made possible by reshaping the dataset. Another term inside a convolutional network is padding. Padding is the pixels amount that is added inside an image at the moment when a kernel is processing it [6]. It can have two values either the same or valid. In this case in our function it is the same [6]. So it means that this convolutional layer may employ zero padding in case that it is needed. The other layer is the Leaky ReLu or in short LeakyReLu. The difference from the ReLu and LeakyReLu is that in LeakyReLU, rather than putting all negative x values to 0, this algorithm puts a small slope which is smaller than 1 for negative values. In the experiment that was conducted, the value of the slope is 0.1, even the output of negative inputs in this function, have a small negative value rather than 0 [21]. The formula of the Leaky ReLU function is given by (19) [12]:

$$Y = 1(x < 0) * (ax) + 1(x \geq 0) * (x) \tag{19}$$

From which Y is the output, the x is the input so in the case that it is negative, the formula is $Y = ax$ and in case when it is positive the formula is $Y = x$ and a is a small constant which will be always smaller than

0. The purpose of this LeakyReLU is to fix the problem of the dying ReLU [21]. The next layer is Maxpooling 2-Dimensional. Maxpooling has the ability to find the maximal value inside a group of numbers. In our experiment every group has 4×4 dimensions. Even in Maxpooling, padding has the value that are the same which means that the Maxpooling Layer is able to employ 0 padding in case that it is needed. These three layers are repeated again in the same way as the first time. The seventh layer in this algorithm is Flatten layer. As the name implies, it is employed to put the inputs one after another on a vertical form or by other words to flatten the inputs. After this layer we have again a LeakyReLU layer with the same slope with 0.1 values. Finally in the output layer, we have a softmax activation function and a number which represents the types of the inputs. The reason why this algorithm was chosen is that the CNN algorithms is able to discover many different DDoS attacks and Malware attacks when the datasets are too large. However, on the other hand, this algorithm takes a very long time to be implemented.

RNN is a Neural Network algorithm that is unlike other neural network algorithms in that these RNNs have both feedforward connections and also feedback connection. This algorithm is employed in anomaly detection. In our experiment, the RNN algorithm has three layers, which in this case are a Long-Short Term Memory (LSTM) input layer, the hidden layer with a sigmoid activation function and the output layer with the softmax activation function. The LSTM [15] is able to control two state vectors and because of performance aims these state vectors are retained unconnected by default. The LSTM contains four different units inside it. The first units which is also the main unit has a general purpose to analyse the contemporary inputs x_t and preceding (in this case short-term) state h_{t-1}. The other three units are also known by the name gate controllers. Because these units employ sigmoid activation functions, they have outputs which vary from 0 to 1. So in this case the units have outputs which if it is 0 (zero), the units will close the gate. On the contrary when the units have outputs of value 1 (one), they will let the gate open. The gate controllers are: forget gate which manages what portions that must be deleted in the long-term state, the input gate which manages what portions must be added in the long-term state and the output gate which manages what long-term portions must be read and the output y_t in this time step. This algorithm is very useful and very easy to be implemented. The LSTM layer is available in the Python library "Keras" [15].

4. Results and Discussions

The results of the training and testing on the 13 algorithms using the four datasets are presented in Tables 1-4, below. The tables give the training times (s) and training accuracies (%) as well as the testing times (s) and testing accuracies (%) achieved. Table 1 presents the results for the kddcup dataset, Table 2 for the nsl-kdd dataset, Table 3 for the Kyoto dataset and finally, Table 4 for the UNSW-NB15 dataset. Analysing the data from all four tables shows that the Gaussian Naïve-Bayes algorithm gives the fastest training and testing times within the range of 0.12 s to 3.41 s, however, with the lowest testing accuracy range of 51-88% compared against the remaining 12 other ML algorithms. The non-linear SVM (2173.8 s) and CNN (870.8 s) were the slowest ML algorithms across the four datasets. Regarding the testing accuracies, if the time factor was excluded, then any of the remaining 12 algorithms except Gaussian Naïve-Bayes exceeded 95% testing accuracies with the four test datasets. For a good balance in terms of overall accuracy and the fastest convergence time, then Random Forest is the best choice to be adopted.

Table 1: Results of training and testing algorithms for kddcup dataset.

Machine Learning Algorithms	Training time (s)	Testing Time (s)	Training Accuracy	Testing Accuracy
Gaussian Naïve-Bayes	2.25	3.41	88.2%	88.2%
Decision Tree	4.92	0.16	99%	99%
Stochastic Gradient	15.5	0.33	99.2%	99.2%
Random Forest	44.64	3.55	99.99%	99.96%
Non-Linear SVM	2173.8	226.42	99.88%	99.87%
Linear SVM	48.26	0.23	99.7%	99.68%
Logistic Regression	40.87	0.23	99.34%	99.3%
Multilevel Perceptron	498.51	0.63	99.41%	99.37%
Gradient Boosting	1433.56	7.99	99.95%	99.92%

Artificial Neural Network	371.18	1.02	98.46%	98.46%
Recurrent Neural Network	448.51	1.1	98.47%	98.47%
Convolutional Neural Network	870.8	21.46	99.86%	99.86%

Table 2: Results for training and testing algorithms for nsl-kdd dataset.

Machine Learning Algorithms	Training Time (s)	Testing Time (s)	Training Accuracy	Testing Accuracy
Gaussian Naïve-Bayes	0.12	0.12	51.29%	51.57%
Decision Tree	0.2	0.016	95.9%	95.86%
Stochastic Gradient	0.62	0.016	97.22%	97.14%
Random Forest	1.18	0.154	100%	99.65%
Non-Linear SVM	3.011	2.04	99.27%	99.06%
Linear SVM	1.4	0.016	96.98%	97.13%
Logistic Regression	1.62	0.015	96.75%	96.69%
Multilevel Perceptron	33.53	0.03	97.48%	97.51%
Gradient Boosting	60.55	0.4	99.96%	99.61%
K-Nearest Neighbour	3.89	21.32	99.49%	99.33%
Artificial Neural Network	192.79	0.74	98.69%	98.5%
Recurrent Neural Network	234.92	0.83	97.83%	97.62%
Convolutional Neural Network	571.16	2.35	99.17%	98.95%

Table 3: Results for training and testing algorithms for Kyoto dataset.

Machine Learning Algorithms	Training Time (s)	Testing Time (s)	Training Accuracy	Testing Accuracy
Gaussian Naïve-Bayes	1.4	0.61	71.14%	71.1%
Decision Tree	3.14	0.078	98.2%	98.19%
Stochastic Gradient	1.96	0.2	94.96%	94.83%
Random Forest	38.47	2.37	99.94%	99.74%
Non-Linear SVM	1958.71	491.59	96.05%	95.88%
Linear SVM	39.5	0.054	94.9%	94.8%
Logistic Regression	10.89	0.046	94.85%	94.78%
Multilevel Perceptron	54	0.2	92.54%	92.54%
Gradient Boosting	173.34	1.06	99.41%	99.35%
K-Nearest Neighbour	259.34	540.64	98.71%	98.19 %
Artificial Neural Network	348.93	1.69	98.45%	98.35%
Recurrent Neural Network	294.88	1.55	98.76%	98.72%
Convolutional Neural Network	703.48	19.54	96.01%	96.06%

Table 4: Result for training and testing algorithm for UNSW-NB15 dataset.

Machine Learning Algorithms	Training Time (s)	Testing Time (s)	Training Accuracy	Testing Accuracy
Gaussian Naïve-Bayes	1.24	0.995	87.94%	86.04%
Decision Tree	2.94	0.1	96.75%	96.81%
Stochastic Gradient	1.49	0.14	95.08%	95.12%
Random Forest	24.6	1.6	99.97%	98.7%
Non-Linear SVM	471.77	308.58	95.45%	94.23%
Linear SVM	9.47	0.116	95.09%	93.88%
Logistic Regression	6.79	0.072	95.07%	94.22%
Multilevel Perceptron	656.31	0.178	97.36%	95.24%

Gradient Boosting	186.66	1.06	98.02%	98.02%
K-Nearest Neighbour	162.62	540.64	97.79%	96.59 %
Artificial Neural Network	206.17	0.51	96.53%	95.76%
Recurrent Neural Network	294.88	1.55	96.72%	96.77%
Convolutional Neural Network	703.48	19.54	95.49%	95.57%

The purpose of these experiments was to show the accuracy and the performance of different types of algorithms that are used to detect intrusions inside a dataset as well as the time taken for the whole dataset to perform detection. By showing the accuracy we need to test them and to make them learn by themselves so in the future they may perform with a higher accuracy in performance.

5. Conclusions and Future Work

In these experiments, different types of machine learning and deep learning algorithms were used to detect different types of malwares inside the datasets. The more the deep learning algorithm were trained, the more precise the results that were obtained. The results of the accuracy of almost all the algorithms that are used for malware detection were relatively good. However, data preparation, in terms of noise removal, filtering and careful selection of the training data for the training of the algorithms need to be carried out to ensure a better performance in terms of achieving higher accuracies. The employment of machine learning algorithm in many different fields of life even in cyber security is the future of the technology. This work investigated 13 different classification algorithms to detect malwares inside four different datasets. 12 of the algorithms provided high accuracies. Algorithms such as the Gaussian Naïve-Bayes classifier, Logistic Regression, Decision Tree classifier take a very short time to be implemented. Except for the Gaussian Naïve-Bayes classifier algorithm whose accuracy is between 51% and 88% the other algorithms have an accuracy over 90%. Algorithms such as Multilevel Perceptron, non-linear SVM and Gradient Boosting take a very long time to be implemented. The algorithm that has performed with the greatest accuracy is the Random Forest Classification algorithm. However, all of them work to provide useful working technology. Here we have built an intelligent system that has a great level of performance. This intelligent system is helpful to discover a lot of cyber threats and it is a very powerful automated system. A future work proposal is to use Reinforcement learning in order to detect intrusions in the system.

References

[1] Sara Najari and Iman Lotfi, "Malware Detection Using Data Mining Techniques", *International Journal of Intelligent Information Systems*, Vol. 3, No. 6-1, December 2014, p. 33-37, DOI: 10.11648/j.ijiis.s.2014030601.16.

[2] Y. Qin and T. Xia, "Sensitivity analysis of ring oscillator based hardware Trojan detection", *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 27-30 October, 2017, Chengdu, China, pp. 1979-1983, ISSN: 2576-7828. DOI: 10.1109/ICCT.2017.8359975.

[3] Douglas Jacobson and Joseph Idziorek, *Computer Security Literacy: Staying Safe in a Digital World*, 1st ed. Florida, USA: Chapman and Hall/CRC, 27 November 2012, ISBN-13: 978-1439856185.

[4] Dipanker Dasgupta, Zahid Akhtar and Sajib Sen, "Machine learning in cybersecurity: a comprehensive survey", *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Vol. 19, No. 1, 19 September 2020, pp. 57-16, DOI: 10.1177/1548512920951275.

[5] Rajashree A. Katole, Swati S. Sherekar and Vilas M. Thakare, "Detection of SQL injection attacks by removing the parameter values of SQL query", *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 19-20 January 2018, Coimbatore, India, pp. 736-741, DOI: 10.1109/ICISC.2018.8398896.

[6] Hafiz M. Farooq and Naif M. Otaibi, "Optimal Machine Learning Algorithms for Cyber Threat Detection", *2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim)*, 27-29 March 2018, Cambridge, UK, pp. 32-37, DOI: 10.1109/UKSim.2018.00018.

[7] Vaishali Bhatia, Shabnam Choudhary and K. R. Ramkumar, "A Comparative Study on Various Intrusion Detection Techniques Using Machine Learning and Neural Network", *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 4-5 June 2020, Noida, India, pp. 232-236, DOI: 10.1109/ICRITO48877.2020.9198008.

[8] Wasim A. Ali, K. N. Manasa, Mohammed Fadhel Aljunid, Malika Bendechache and P. Sandhya, "A Review of Current Machine Learning Approaches for Anomaly Detection in Network Traffic", *Journal of Telecommunications and the Digital Economy*, Vol. 8, No. 4, 2020, pp. 64–95, Online ISSN 2203-1693. DOI: 10.18080/JTDE.V8N4.307.

[9] Mahesh V. Chari and Sumithra Devi K.A., "Prevention from Security Risks of Spyware by the use of AI", *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 19-20 March 2019, Bangalore, India, pp. 131-135, DOI: 10.1109/ICATIECE45860.2019.9063838.

[10] I. Sumantra and S. Indira Gandhi, "DDoS attack Detection and Mitigation in Software Defined Networks", *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 3-4 July 2020, Pondicherry, India, pp. 1-5, DOI: 10.1109/ICSCAN49426.2020.9262408.

[11] Priyanka Dixit and Sanjay Silakari, "Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review", *Computer Science Review*, Vol. 39, 2021, 100317, ISSN 1574-0137. DOI: 10.1016/j.cosrev.2020.100317.

[12] Prajakta M. Ombase, Nayana P. Kulkarni, Sudhir T. Bagade and Amrapali V. Mhaisgawali, "DoS attack mitigation using rule based and anomaly based techniques in software defined networking", *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 23-24 November 2017, Coimbatore, India, pp. 469-475, DOI: 10.1109/ICICI.2017.8365396.

[13] Rabie A. Ramadan and Kusum Yadav, "A Novel Hybrid Intrusion Detection System (IDS) for the Detection of Internet of Things (IoT) Network Attacks", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 61-74, Vol. 4, No. 5, 20th December 2020, DOI: 10.33166/AETiC.2020.05.004, Available: http://aetic.theiaer.org/archive/v4/v4n5/p4.html.

[14] Bavhani Thuraisingham, "The Role of Artificial Intelligence and Cyber Security for Social Media", *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 18-22 May 2020, New Orleans, LA, USA, pp. 1-3, DOI: 10.1109/IPDPSW50202.2020.00184.

[15] Aurélien Géron, *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*, 2nd ed. California, USA: O'Reilly Media Inc., 15 October, 2019, ISBN-13: 978-1492032649.

[16] Shree Krishna Sharma and Xianbin Wang, "Toward Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions", in *IEEE Communications Surveys & Tutorials*, Vol. 22, No. 1, pp. 426-471, Firstquarter 2020, DOI: 10.1109/COMST.2019.2916177.

[17] Yaping Chang, Wei Li and Zhongming Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine", *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 21-24 July 2017, Guangzhou, China, pp. 635-638, DOI: 10.1109/CSE-EUC.2017.118.

[18] M. A. Jabbar, Rajanikanth Aluvalu and S. Sai Satyanarayana Reddy, "Intrusion Detection System Using Bayesian Network and Feature Subset Selection", *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 14-16 December 2017, Coimbatore, India, pp. 1-5, DOI: 10.1109/ICCIC.2017.8524381.

[19] Altyeb Altaher, "Phishing Websites Classification using Hybrid SVM and KNN Approach", *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 6, 2017, pp. 90-95, DOI: 10.14569/ijacsa.2017.080611.

[20] Ishita Saha, Dhiman Sarma, Rana Joyti Chakma, Mohammad Nazmul Alam, Asma Sultana and Sohrab Hossain, "Phishing Attacks Detection using Deep Learning Approach", *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 20-22 August 2020, Tirunelveli, India, pp. 1180-1185, DOI: 10.1109/ICSSIT48917.2020.9214132.

[21] Kinam Park, Youngrok Song and Yun-Gyung Cheong, "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm", *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, 26-29 March 2018, Bamberg, Germany, pp. 282-286, DOI: 10.1109/BigDataService.2018.00050.

[22] David G. Kleinbaum and Mitchel Klein, *Logistic Regression A Self-Learning Text*, 3rd ed. Heidelberg, Germany: Springer, July 2010, ISBN-13: 978-1441917416.

[23] Michal Kedziora, Paulina Gawin, Michal Szczepanik and Ireneusz Jozwiak, "Malware Detection Using Machine Learning Algorithms and Reverse Engineering of Android Java Code", *International Journal of Network Security & Its Applications (IJNSA)*, Vol. 11, No. 1, January 2019, pp. 1–14, DOI: 10.5121/ijnsa.2019.11101.