WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Novel Cyber Security Intrusion Detection Model based on Machine Learning

[1]FAGBOHUNMI, Griffin Siji, [2]UCHEGBU Chinenye E.

[1]Computer Engineering Department Abia State University, Uturu, Abia State, Nigeria. fagbhume, griffn@abiastateuniversity.edu.ng
[2]Department of Electrical and Electronic Engineering Abia State University, UturuAbia State Nigeria , ceuche@gmail.com

ABSTRACT:

The attention received by Cyber security systems in recent times has been high due to the continued proliferation of attacks and threats ever present on the internet through cyber-attacks on systems used in establishments, such as banks, industries and educational institutions. This has also led to the increased popularity of the Internet of Things (IoT), (as this constitutes the majority of devices sharing information on the internet with one another), rapid development of computer networks and increase in cyber security applications. The increase in cyber security applications also contributes to possible detection of unusual events, traffic and cyber-attacks on the network, thereby leading to the design of a reliable intrusion detection system capable of identifying weakness in these adversarial threats. Research has shown that machine learning, which is a subset of artificial intelligence may be employed to design an intelligent intrusion detection system. It is therefore the aim of this paper to design a novel machine learning based intrusion detection system using binary tree. (BTIntruDS). This will be achieved by the intelligent agent identifying the security features in a system based on its relative importance in defending the system from being intruded. A binary tree data structure will then be used to develop an overall intrusion detection model where the most important security feature will be at the root of the tree. The model will be able to predict accurately the nature of any attempted access to the system, similarly, the software implementation will also not be computationally complex. The BTIntruDS model was tested by carrying out cybersecurity experiments and using the security dataset from the experimental setup to compute the recall, pscore accuracy, precision and ROC values in order to determine the efficiency of the model. The results from the experiment was compared with other machine learning methods such as, naïve Bayes classifier, logistic regression, support vector machines and k-nearest neighbour to compute the relative effectiveness of the BTIntruDS model with these models. BTIntruDS outperforms the compared protocol by an average of 13%.

Keywords: cybersecurity; cyber-attacks; anomaly detection; intrusion detection system; machine learning; network behaviour analysis; cyber decision making; cybersecurity analytics; cyber threat intelligence

1. Introduction

Nowadays the search for software that can provide cybersecurity and protection of endpoint devices from different types of cyber-attacks has been on the increase. This can be attributed to the increased proliferation of attacks and threats ever present on the internet through cyber-attacks on systems used in establishments, such as banks, industries and educational institutions, this has also led to the increased popularity of Internet of Things (IoTs). The evolution of cyber-attacks in computer networks and the diverse network applications designed for different categories of endpoint devices connected to a network has also led to the continued interest by researchers in finding ways to ameliorate this menace. In literature, the different types of cyber-attacks like the denial of service (DSS) attack (Sun et al 2010), computer viruses (Dainotti et al 2015), has resulted in devastating consequences to assets of various companies and banks with the attendant loss in finance especially where such network covers extensive geographical space. According to the researchers in (Qu et al 2022), in September 2023, a particular virus resulted in huge financial losses for many companies and organizations in Nigeria. The companies span across banks, educational institutions, Internet service providers just to mention but a few. The financial loss ran into several billions of Naira. A recent survey shows that most multinational companies that suffer network data intrusion loses between 20 – 35 billion Naira (IBM Security Report. 2022). These occurrences has led to an increase in the demand for software capable of providing cyber-security protection to a network. It should be noted that there has been an increase on daily basis in the different types of cyber-attacks on company's network. The reason for these are not far-fetched. The perpetuators of this act gain a lot financially and that has been the drive towards such behaviour.

In a normal setting, a cyber-security system is made up of (i) a network security system and (ii) a computer security system. It should be noted that additional features like an encryption system and firewall can be added to the security functionality to protect end to end communication and defence against unknown attack and intrusion respectively especially when the network are online.. In order to protect a computer network from external attacks, it is necessary to design and build an intrusion detection system (IDS) (Tsai et al 2019). In order words the purpose of an intrusion detection system is to identify any dangerous communication within the network and also prevent adversarial agents capable of network intrusion from gaining access to the network. It should be noted that the conventional methods of firewalls are incapable to perform this function (Mohammadi et al 2023), (Tapiador et al 2018), (Tavallaee et al 2017). The features in an IDS comprises of being able to identify agent behaviour that can be classified as

malicious on the network and taking appropriate actions to prevent it. An IDS should also be able to keep log of the daily activities on the network so as to build an information base of what actions are possible or not on a given network. Using an instance of a computer network, that has a built in functionality of detecting security breach or threat such as denial of service (DoS), such system must have the capability of identifying what constitutes unauthorized behaviour and this can only be detected when daily log of activities in the network are kept. The system should also be able to identify unauthorized access and prevent it so as to protect the network's data from being modified and corrupted (Milenkoski et al 2020), (Buczak and Guvenb2021), Xin et al 2022). It should be realized here that if unauthorized access and abnormal behaviour is adequately monitored successfully on the network, its data will be free from being modified or corrupted. It is therefore important that an effective IDS must be designed to protect a network from various types of cyber-attacks and abnormal behaviour on the network.

There are various types of intrusion detection system depending on the scope of its usage. The most used types of intrusion detection systems are the host-based and network based. This can be implemented on single computers and large networks respectively Moskovitch et al 2014). In the case of the host based intrusion detection system, it works on a single system and it monitors some files in the operating system whose infection can be catastrophic to the system. It should be noted here that a host intrusion detection system (HIDS) effectiveness is limited to only a system, or a very small set of systems. However in the case of the network intrusion detection system (NIDS), the software agent monitors the network system for any unusual type of traffic. This is done by the system having a log of the data communication on the network on a daily basis, which is then used to form a basis for the network communication pattern. The intelligent agent in the network system will be able to detect abnormal behaviour on the network by correlating the current situation to what has been stored in its log database. The common technique used are variants of signature based and anomaly based detection. These two techniques have been studied extensively in literature for quite sometimes (Sommer,and Paxson 2017). In the case of the signature based detection system, a unique pattern will be used to detect suspecting threats. In signature based detection system, a sequence of byte in a network communication which is considered unusual or an already detected sequence of bytes used by a malware may be used as a signature. This types of patterns are used as signature in antivirus software. This is normally done by matching the pattern in the signature with the incoming data bytes, and if a correlation is found a malware detection flag will be raised. It should be noted hare that the signature based detection method can only detect already known threats, and may be hard to detect new malware whose byte pattern is not yet known. However, in the case of the anomaly based intrusion detection system, the nature of the network pattern is examined, and a profiling of a data-driven model of the network communication is done to classify normal and abnormal pattern. In this way, there will be a margin between a normal network communication pattern and abnormal network communication pattern. This helps the anomaly based intrusion detection system to detect unknown malware.

The ability to detect yet to be identified or unknown malware gives the anomaly based intrusion detection system an edge over the signature based intrusion detection system. The shortcoming of the anomaly based intrusion detection system is its high false alarm rate. This is used to describe a situation where the detection system signals a normal network traffic as malicious [10]. This brought about the need for a machine-learning based detection technique that will overcome the shortcoming of the anomaly based IDS.

It is on this backdrop that a novel machine learning based intrusion detection system is developed in this paper. The machine learning based IDS will reduce the rate of false alarm usually witnessed with the anomaly based IDS. The model proposed in this paper will contribute to designing a relatively less memory and computationally intensive algorithm to analyse various cyber threat patterns and predict those incidence of adversarial attacks on the cybersecurity data. This will help in the development of an intelligent data driven intrusion detection system. The use of machine learning for this design is built upon the precept that intelligent agents are capable of learning from their environment especially using supervised learning. Hence this feature will assist the model to learn from dataset of known and projected threats (Seufert and O'Brien, 2014), in this model a binary tree will be used as the data structure for the model, this is due to its well-known good performance in predictive analysis (Sarker et al 2023), (Sinclair et al 2012). Modelling cyber-attacks in an effective way is quite challenging from an analysis of threat datasets in current cyber security system. This is because the security features in threat datasets may have high proportions of unimportant or less important information, which may impact the correct prediction of the status of a program. Threat dataset having this characteristics may result in dataset analysis having high variance resulting in over fitting in a tree based structure. This is because the model may only learn from a minimal subset of the tree resulting in high computation cost and time consumption for learning. This is because the entire tree structure is traversed for information contained in only a minimal subset of the security datasets. It may also cause the model not to define the security pattern adequately which may ultimately cause reduced accurate prediction of threat patterns especially for unknown threats.

For this reason, a binary tree intrusion detection system (BTIntruDS) based on machine learning is proposed. The system aims to minimize the shortcomings in earlier systems designed for the same purpose. In BTIntruDS, the security features in the system will be modelled in such a way that the most important security model will be designated as the root node. A binary tree is then built such that the position of a security feature will be in a top down layer based on decreasing importance of the security feature. The full binary tree is built when the intelligent agent in the model has been adequately trained on the order of importance of the security features based on the security data so that its position will be properly placed on the binary tree. Before the final position of a security feature is determined on the binary tree, a test data will be used to authenticate the model. The model is therefore not computationally complex as the feature dimension is reduced and also it helps in reducing over-fitting in modelling due to incorrect prediction. It should be noted that the use of test data will enhance prediction accuracy on yet to be identified test cases.

In general, the contribution made in this work are stated as follows: (i) the importance of security features are highlighted with particular reference to high dimension security features in machine language based intrusion detection system. (ii) A binary tree intrusion detection system 'BTIntruDS' based on machine learning security model is built in a top-down ranking based on the decreasing importance of the security feature. (iii) reducing security issues in the building of an IDS by using a subset of the security dataset (iv) design a data-driven intrusion system that is effective in predicting threat pattern in a cybersecurity system (v) An experimental test bed is used to validate the relative advantage of the BTIntruDS model over other intrusion models. From the experiments conducted, the BTIntruDS model showed significant increase in performance in the detection of online / cyber intrusions when compared with other state of the art models used for comparison especially for yet to be identified test cases..

The remaining part of this paper is as follows: the second section provides related works based on contributions of other researchers in this area of study, the third section presents the machine learning model for the BTIntruDS binary tree model based on the relative importance the security features

in the intrusion detection pattern. The forth section describes the results from the analysis conducted based on the cybersecurity dataset from the experimental test-bed. The fifth section concludes the paper and gives areas of further research directions.

2. Related Works

The main purpose of an intrusion detection system is to detect program sequence that results in adversarial cyber-attack on a network and at the same time keeping track of daily occurrences on the network in order to identify adversarial threats on it (Milenkoski et al 2020), Xin et al 2022). There has been a plethora of research work done in the area of cybersecurity with the sole aim of identifying and forestalling security breaches, intrusion or cyber-attacks. In recent times, the signature based network intrusion system has been used extensively in cyber security (Seufert and O'Brien,2014). In the system, a known signature of already identified threat is used for the detection of other yet to be identified threats. This method has been used widely in the industry to tackle cyber security concerns and as witnessed success commercially in recent times. Also the anomaly based intrusion detection system has been employed in more recent times. The anomaly based method has an edge over the signature based approach because it can better identify yet to be known threats (Alazab et al 2020). The anomaly based approach tracks network traffic and identifies attack pattern by evaluating the pattern for the security data. The signature based approach is only very good in identifying already known threats and its variants. It cannot successfully detect a new threat using a completely unknown signature pattern. There are different data mining and machine learning based methods used in analysing data with incident pattern in security concern for making useful decision on the status of a given data (Sarker et al 2023), (Han et al 2018), by this we mean data pattern that is susceptible to security breaches. The shortcoming of the anomaly based intrusion detection technique is that it gives high rate of false alarms this means classifying yet to be identified trustworthy data pattern as threat and vice versa (Buczak and Guven 2021). It is therefore pertinent that a model capable of minimizing rates of false alarm must be sought (Sommer and Paxson 2017). It has been found that machine learning may present a technique through which an efficient intrusion detection mechanism can be deployed to minimize that instance of false alarms. The sub-area of machine learning where this effective intrusion technique can be designed has to do with data mining and computational statistics where intelligent agents are made to learn from data patterns (Han et al 2018), (Witten and Frank 2014). This area of machine learning also has close relation to other scientific area such as optimization and mathematical theories. It is therefore pertinent that scientists in area of cybersecurity must first be able to analyse the underlying security data pattern as the technique is data driven. This will be needed to design a security model that is intelligent to accurately predict security data patterns for yet to be identified security patterns. Even though association analysis is widely used for the design of rule-based intelligent systems in machine learning techniques (Agrawal and Srikant 2008), (Sarker and Salim 2023), (Sarker 2023). In this paper, the attention will be based on classification learning techniques.(Sarker 2022). Classification learning techniques are widely used in the design of predictive models that relies on an underlying dataset training. There are different methods for designing a data-driven predictive model, they include naïve Bayes classifier, hyperplane-based support vector machines, instance learning, k-nearest neighbour, logistic regression technique, sigmoid function and the method used in this paper, rule-based classification i.e. decision tree (Sarker et al 2023), (Han et al 2018).

The researchers in Li et al 2019), classified predefined attack categories such as Denial of Service attack (DoS), U2R, Probe or scan, R21, together with the normal traffic by using the KDD'99 cup dataset. This was based on the hyperplane based support vector machine with RBF kernel. According to the researchers in (Amiri et al 2019), a large dataset was trained using the least-squared support vector machine classifier in order to develop a faster training system. According to the researchers in (Hu et 2017), a variation of support vector machine classifier was used to classify anomaly pattern. According to the researchers in (Wagner et al 2019), they designed a classifier based on one-class support vector machine for the purpose of detecting anomaly patterns in various types of attacks, which includes NetBIOS scans, POP spams, DoS attacks and secure shell scans. The researchers in [29], employed the same support vector machine classifier used by the researchers in (Wagner et al 2019), in order to detect characteristics of yet to be known viruses and worms. The same one-class support vector machine classifier was employed by several other researchers as found in the works in (Kotpalliwar and Wajgi 2020),for designing an intrusion detection system.

It should also be stated here that many other types of classifiers have been used in literature for the purpose of intrusion detection. For example in the work of researchers in (Kruegel et al 2016), they employed the probability based Bayesian network to identify different events that constitute the process in TCP/IP packets. Also in the work of authors in (Benferhat et al 2013), the Bayesian network was also used for the detection of Denial of Service (DoS) attack. In the work of authors in (Panda and Patra 2017), the naïve based Bayes probability classifier was used in the design of the analysis of the KDD'99 cup data sets, wherein four different types of attacks were analysed which comprises of DoS, Probe or scan,R2L and U2R. In the work of researchers in (Koc et al 2020), a multi-class intrusion detection system was designed using the naïve Bayes classifier. Researchers in (Vishwakarma et al 2021), used KNN which is an instance based learning algorithm for classifying different points which location depends on the K-nearest neighbours of data points for intrusion detection. The authors in (Bapat et al 2022), (Besharati et al 2023) employed the logistic regression model for the identification of dangerous traffic and data intrusion. Several other techniques such as the neural classifier has also been employed by researchers in (Kumar and Selvakumar 2018), while researchers in (Dainotti et al 2017) employed the wavelet transform for the purpose of anomaly detection especially in DoS arracks.

A tree based machine learning is among the most used classifier technique in the design of predictive models. Predictive models in machine earning are designed to efficiently and correctly predict the pattern of anomaly behaviour or pattern in network traffic, and as such will be very ideal in classifying yet to be known virus attack or worms. Some of the best used design of decision trees includes the C4.5, (Quinlan 1998), ID3, (Quinlan 2009) algorithms. In the work of researchers in (Sarker et al 2022), a behavioural decision tree algorithm known as BehavDT was designed for the analysis of behavioural pattern. A very good number of research work such as in (Ingre et al 2021), Iqbal et al 2023) employed the decision support trees classifier technique for the design of an intrusion detection system, but their shortcoming was the high dimension of security issues. This may result in high variance in the pattern classification which may result in non-convergence of the classifier pattern. This will eventually result in over-fitting in dataset values and huge complexity in computation time and cost, which may ultimately result in low accuracy of prediction.

The work presented in this paper, BTIntruDS provides sorting of security features in the system in descending order of importance. A binary tree is then built in a top to bottom classification method to determine the position nodes) corresponding to a particular security feature.

3.1 Materials and Methods

This section presents the model of the binary tree based intrusion detection system (BTInrDS) proposed in this paper. The model is made up of a number of stages which explore the security dataset in a network. From the security dataset, raw data are extracted which assists in assigning the feature importance used in assessing the position of the dataset in the binary tree model. The following subsections describes the stages that make up the model.

3.2 Security Dataset Exploration

A cybersecurity intrusion detection model is made up of security datasets contained in the information database. This information database are used in determining the security features and other accompanying details required in building a data-driven cybersecurity intrusion detection system. It is therefore pertinent to know the properties of raw cybersecurity data as well as security incident patterns in order to extract important information from them.

In this paper, an intrusion detection system based on three types of variables was considered, this includes, normal, irrelevant and abnormal. The dataset used comprises of 43 features, among which 3 features were qualitative while the other 40 features were qualitative, this includes range of time logged in, an error state, data host count, number of bytes in security feature. The full complement of the security features are shown in table 1. From this table it can be seen that there were over thirty thousand threat instances compiled from many intrusion that were simulated on a bank's network. In order to gather the raw data, which includes TCP/IP and NETBIOS dump data in the bank's network. The network topology was simulated using a well-known Nigerian bank local area network. The network was developed in such a way as to assume that the bank's environment was a target to many adversarial/malicious hackers. Many adversarial threats were then incident on the network.

Table 1 Datatype of the selected Dataset features

Feature Name	Data Type	Feature Name	Data Type
dist_host_ser.count	Integer	equal_srv_rate	Float
Flag	Nominal	dist_host_equal_server_rate	Float
serving_server_rate	Float	dist_host_server_error_rate	Float
dist_host server_rate	Float	Count	Integer
protocol_type	Nominal	log_in	Integer
dist_host_equal_server_port_rate	Float	dist_host_server_diff_host_rate	Float
nerror_rate	Float	srv_bytes	Integer
dist_host_server_error_rate	Float	Service	Nominal
srv_error_rate	Float	dist_host_pserror_rate	Float
dist_host_count	Integer	dist_host_diff _server_rate	Float
server_count	Integer	correct_ f ragment	Integer
pserror_rate	Float	num_attacked	Integer
server_diff _host_rate	Float	dist_bytes	Integer
Host	Integer	diff _server_rate	Float
time_limit	Integer	out_login	Integer
root_shell	Integer	Kand	Integer
real_time	Integer	num_ unsucessful_logins	Integer
di_attempted	Integer	num_root	Integer
number_ f ile_creations	Integer	num_branches	Integer
number_access_ f iles	Integer	num_outbound_cmds	Integer
in_host_login	Integer		

It should be noted here that the security features shown in table 1 above are expected to have different probability distribution. In order to buttress this, figure 1 and figure 2 show the probability distribution for two different security features, together with the time limit and the number of bytes of the threat feature. The raw dataset of the security feature values were first determined from the values assigned to the threat features in order to build the binary tree based intrusion detection model. It is necessary to be able to rank security features outlined in table 1 in accordance to the desired parameters required to build the intrusion detection model that is based on a data driven pattern. The correct ranking of these security features will assist in accurate prediction of almost all yet to be identified threats. Correct prediction of impeding intrusion will go a long way in protecting a network from various from of known and unknown adversarial threats.

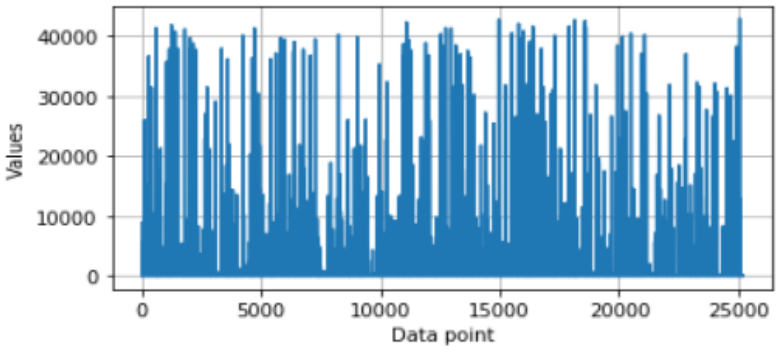

Figure 1. Value distribution of the security feature 'time-lag'.

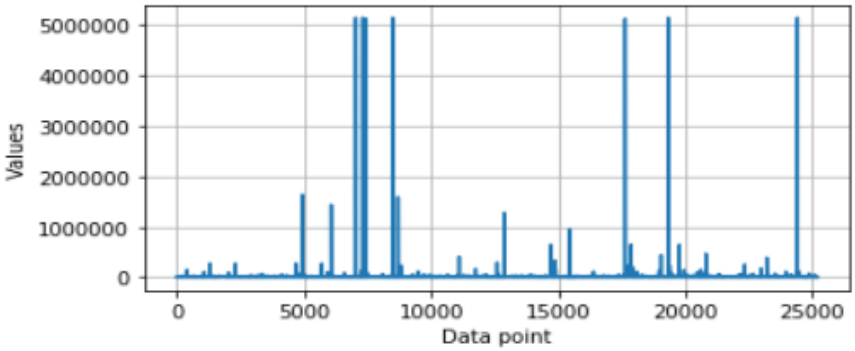

Figure 2. Value distribution of the security feature 'data bytes'.

3.3 Raw Security Data Preparation

In order to prepare data gotten from an assumed intrusion dataset, two processes are involved, these include, feature encoding and scaling. In order to perform feature encoding, it should be realized that any intrusion dataset contains numeric and nominal values. Even though most feature value from an intrusion dataset are inherently numeric, some are still nominal, i.e. service status flag, protocol type, and class type which can either be normal or abnormal just to mention but a few. The basic requirement is to convert all nominal feature values to vectors so that it can be matched to the built intrusion detection model.

Label encoding is used in this paper instead of one hot encoding (as these are the two main encoding techniques used in machine learning for encoding variables). This is because unlike one hot encoding, label encoding can easily convert feature values to numeric value which is important in analysing the model for extracting useful information from the dataset. Also one hot encoding usually results in an increase in feature dimension especially as the size of dataset increases.

Another important operation in intrusion detection is feature scaling. This is an act of data pre-processing in which the values of security features are normalized so as to have uniform encoding for security features with varying sizes. As stated earlier figure 1 and figure 2 shows graphs which represent the distribution of data from two separate security features depicting range of time and size of data in bytes respectively. It can be noticed from the graphs that the values of data points vary largely among the various security features. Feature scaling is therefore used to set the range of the different data values (normalization) from the security features to be in the same range irrespective of their initial ranges.
This was accomplished by using a standard scaler with a mean value of 0 and variance of 1 to normalize the security feature notwithstanding the initial feature range. This normalized values can then be substituted into the intrusion detection model for further analysis.

3.4 Computing Ranking and Feature Importance

In the model proposed in this paper, the data pattern intending to access the network will be explored and accessed, data will then be separated into normal and abnormal. The importance value of individual abnormal security feature will then be computed. The position of the abnormal security feature will then be decided by its value. It should be noted here that an abnormal security feature with the highest value (importance) in the security dataset will be placed at the root of the binary tree. The rest will then be placed according to their values in descending order on the two-child branches of the root. This process is continued until all abnormal security features have been placed on the binary tree. From the aforementioned, it will be noticed that the higher a node is on the binary tree the higher the threat influence of the security feature, which translates to the fact that such security features will corrupt more folders of data on the network. The idea here is to devise a means of reducing the probability of certain security feature

intruding the network. The probability of an abnormal security feature on a particular node in the binary tree is computed from its number of repetitions on that node divided by the total number of attempts by such security feature. The value range of this probability is from 0 to 1. When the value is 0, it means that the model output is not dependent on the feature, while a probability of 1 implies that the model output is strongly linked with the feature. On this premise, the degree of impurity of any security feature is defined by the probability by which its status will not be wrongly defined. This means that a security feature with a high degree of impurity means that it has a high probability of being wrongly defined and vice-versa

Gini Index is used in data mining to compute a node's impurity. It is a computational model used in determining the probability that a random element will be wrongly classified in a dataset class distribution [19]. If the classification split is binary, then the Gini Index is as shown in equation 1.

$$G_I(k) = 1 - \sum_{m=1}^{D} r_m^2 \quad (1)$$

$$\Delta G_I(k_r) = G_I(k_r) - r_l G_I(k_m) - r_n G_I(k_n) \quad (2)$$

Here r_m depicts the probability that a given element is represented under a certain security class and r_l and r_n are fractions of the samples in node k_i which are allocated to child nodes k_m and k_n respectively. The rate at which the impurity is decreased by a given feature is calculated from the Gini impurity formula given in equations 1 and 2. It should be noted that the importance of any given feature is defined by how much it decreases the impurity i.e. wrong classification. A feature that decreases the impurity with a higher rate will have higher importance value than that that decreases the impurity with a lower rate. In the model, the position of each security feature in the binary tree goes down progressively as its computed security score decreases. After the values of the impurity of the security feature is computed, it is then assigned to the appropriate node in the binary tree. In assigning the security features to nodes in the binary tree it should be noted that each of the two child nodes of the non-leaf node in the binary tree must have impurity values close to each other, otherwise the such non-leaf node will have a single child node. The model is made in this way so as to easily apply the binary search algorithm in detecting any given security feature. This behaviour of the binary tree makes it easy to detect pattern and correlations in a given security dataset. The arrangement of the security features in the binary tree will also enhance the reduction of the dataset to a lower dimension without any significant loss of information.

3.5 Intrusion Detection Tree Design

The binary tree model for the intrusion detection system will be designed after when the security features are processed. The binary tree is required so that informed decisions can be taken in an intrusion detection system that is data driven in nature, As it was stated in the section 3.4, the arrangement of the security features on the binary tree is top bottom according to the value and ranking of their importance (importance score). Therefore in the design only those security features which have high rank or importance score are represented on the binary tree. In order words not all security features in a given security dataset will be represented on the binary tree model. The security feature with the highest importance score is placed as the root. The training dataset is then broken down into smaller datasets with the highest ranking security feature placed at the root of the binary tree. The rest of the training dataset are then grouped in pairs (under each parent node) according to their importance score, in such a way that security features being grouped as two-child nodes under a given parent node will have close importance score. In a situation where the two-child security feature under a given parent node doesn't have a security feature with a close importance score, it will be placed singly under its next higher node (parent node). The essence here is to be able to use the binary search algorithm to easily locate the position of a security feature on the tree. The binary search algorithm will be needed when it is necessary to change the location of a security feature on the binary tree if its importance score suddenly change in the future.

The arrangement of the security features is in a binary tree structure for easy analysis. The height of the binary tree is also dependent on the selected security features from the training dataset. The security dataset is divided into normal and abnormal as depicted in figure 3. It should be noted here that regular security check updates are done on the training datasets as new security features can be identified with time while some may fall off the pecking order in the binary tree and thus discarded. The regular security feature update is used to keep the intrusion detection system current with time and relevant as time progresses. The BTIntruDS model designed in this paper has three inherent properties, (i) Decreasing the size of the security feature selected from the training dataset using the ranking and the importance score. (ii) Design a binary tree whose height is dependent on the selected security features from the training dataset, (iii) Regular update of the training dataset to keep track of changing importance score of any security feature as well as new datasets. An example of the BTIntruDS is shown in figure 3 illustrating some parameters such as status flag, service, duration of intrusion, log in time of intrusion together with the computed values from the training dataset.

The steps required in the construction of the BTIntruDS is shown in Algorithm 1. If a training dataset DATA = $\{Y_1, Y_2, \ldots\ldots Y_n\}$, where n depicts the size of the data, Every instance is defined by m-dimensional matrix. The training dataset is assumed to be from various cyber-attack classes CAC = {normal, abnormal}. The result of the BTIntruDS will be classified as rule based binary tree having a relation with CAC. As an example from figure 3, a single rule feature may be if the flag value is FSTR, the result will be abnormal. On the other hand, a multiple rule feature may be the 'if flag value SD, service is dtb and duration <=5, in this case, the result will be abnormal'. In other words by traversing the constructed BTIntruDS, a range of security rules can be extracted. The outcome of this traversing will help to determine if an intrusion test case is normal or abnormal.

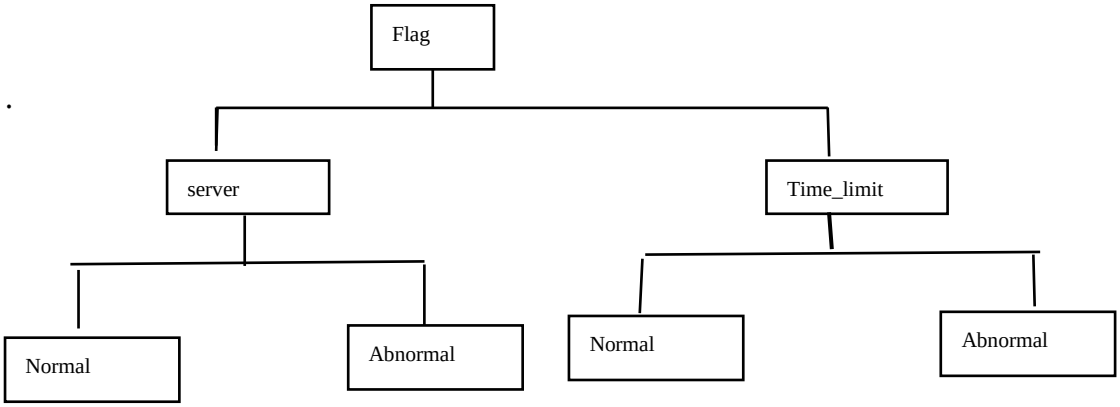

Figure 3. Example of Binary Tree for some features in BTIntruDS

Algorithm 1: BTIntuDS Induction
Data : Dataset: DATA = Y1, Y2, ..., Yn // each dataset Yi contains a number of features and accompanied cyber-attack class CAC
Output: A BTIntruDS Tree construction
1 Procedure BTIntruDS (DATA, feature_list, CACs);
2 // compute feature importance score
3 imporscore ← compute Score(feature_list)
4 / /select important features
5 impor_ feature_list ← selectFeatrues(feature_list, impor_score, m)
6 BTreeGen(DATA, impor_feature_list, CACs)
7 M ← createNode() // create a root node for the tree
8 if impor_score M_k= highest
9 M_k ← rootNode
10 sort impor_score in descending order $M_0 – M_{k-1}$
11 for n = 0 to k-1
12 set M_0 and M_1 direct child nodes to rootNode defining binary tree
11 if $M_2 \sim M_3$ .AND $M_4 \sim M_5$
12 then M_2, M_3 are direct child node to M0 and M_4, M_5 are direct chid node of M_1
13 else M_2 is lone child node of M_0 and M_3 is direct child node of M_2
14 if all instances in DATA belong to the same class CAC then
15 end if
16 return M_{k-1} as a leaf node labelled with the class CAC.
17 end for
18 if impor_feature_list is empty then
19 return M_1 as a leaf node labelled with the majority class in DATA; // majority voting
20 end
21 find the highest precedence feature Fdiv for dividing and assign Fdiv to the node M.
22 for each feature value val ∈ Fdiv do
23 create subset $DATA_{sub}$ of DATA containing val.
24 if DS_{sub} ! = φ then
25 add the node returned by TreeGen($DATA_{sub}$, {impor_ feature_list − Fdiv}, CACs)) to node N;
26 end
27 attach a leaf labelled with the majority class in DATA to node M;
28 end
29 return M

Figure 4. Algorithm for Induction steps in BTIntruDS

4 Results and Analysis of Experiment

This section discusses the result and analysis from the experiments conducted using the BTIntruDS model. The first part of the analysis involves an evaluation of the BTIntruDS cyber security model, the result for other datasets are then analysed.

4.1 Setup of the Experiment

The following criteria will be used to study the BTIntruDS model proposed in this paper: Firstly, the feature importance and ranking used must reduce redundancies in the datasets and present a simplified and manageable datasets to the model. This is aimed at constructing a model that is data-driven such that the properties of any intrusion attempt can be described by a mathematical model. Secondly the BTIntruDS must be capable of correctly detecting both known and yet to be identified cyber intrusion using the data driven model, so that new data intrusion attempts can be prevented. Thirdly it must be able to compare the model proposed in this paper to other machine learning based intrusion detection methods, and finally the model must be updatable, so that it can be easily adaptable to new security dataset trends.

In order to satisfy these criteria, experiments were conducted on security datasets comprising both normal and abnormal dataset classes as stated in section 3.4. All the datasets classes were implemented in C++ program. Scikit-learn, a machine learning library was used for the prediction of the security dataset class. The next section provides a definition of the evaluation metrics used for the BTIntruDS model, after which the results from the different experiments carried out was provided.

4.2 Evaluation Metrics

The following metrics were used to analyse the comparative advantage of the BTIntruDS model proposed in this paper. They are precision, recall, pscore, ROC value and total accuracy, this are given by the following equations

$$\text{Precision} = \frac{CP}{CP+IP} \quad (3)$$

$$\text{Recall} = \frac{CP}{CP+IN} \quad (4)$$

$$\text{Pscore} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

$$\text{Accuracy} = \frac{CP+CN}{CP+CN+IP+IN} \quad (6)$$

From equation 3 to 6, CP represents correct positives, IP represents incorrect positives, CN represents correct negative, and IN represents incorrect negatives. A comparative metric referred to as the ROC (Receiver Operating Characteristics) curve was used to evaluate the ratio of the measured positives with the measured negatives. This was obtained by plotting the rate of correct positives (RCP) versus the rate of the wrong positives (RWP) from the obtained security model.

4.3 Feature Importance Score and Ranking Effect

The feature importance effect was measured by comparing the importance score of each dataset feature with the significance of the score. The significance of a score is determined by the number of times such score is repeated in the dataset. The feature importance score as computed in the BTIntruDS is shown in figure 5. The figure shows the computed importance score for the different features selected from the security dataset. From figure 5, it can be deduced that from a given dataset, the computed importance score are not similar for all selected features. It is different for the various selected security feature. Their effect on the intrusion model is determined by how much influence they have on the security dataset. From figure 5, it can be seen that the data host service count feature is the security feature with the highest value. Its value is 0.256 constituting about a quarter of the entire security dataset. On the other hand, the value of the server host rate security feature which defines when the system is being intruded by a cyber threat is very close to zero, a phenomenon that shows that the system is highly secured against cyber threats. The importance value for the selected security features are arranged in descending order so as to highlight the importance of each security feature in the dataset. This helps one to view the BTIntruDS model of each security features in terms of its influence on the entire datasets

It can therefore be concluded that since the security features has different importance value, it will be wise to exclude the security features with very small importance value (i.e. IV = 0.05 and below) from the dataset in the model. This will help simplify the BTItruDS model as it will reduce the model's complexity. If this value of importance value is used, it will reduce the number of selected features to only 14 from the available 42. The selected security feature is shown in table 2. It can therefore be

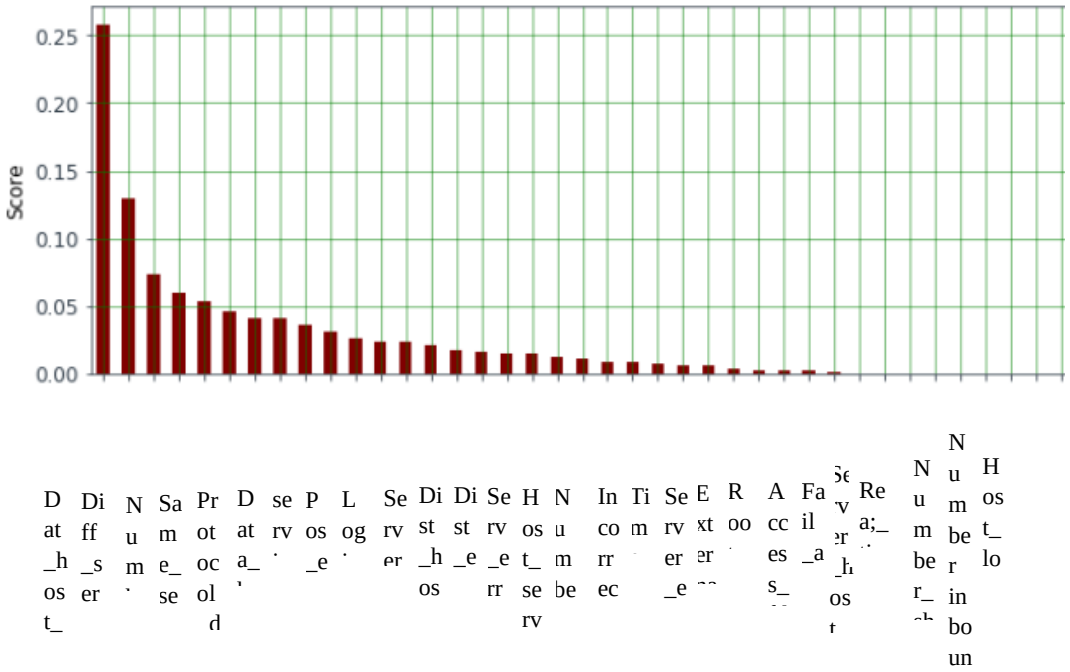

Figure 5. Security importance features with score values.arranged in descending order

deduced that the ranking procedure of the security features together the appropriate pruning of security feature with low value will aid in reducing the complexity of the BTIntruDS model. This means that a general model of a data-driven security model can be built from a reduced number of security features.

Table 2. Importance Value of theTop 14 ranked features using the selected security dataset.

Rank Importance	Name of Security Feature	Value
1	src_bytes	0.258193
2	data_bytes	0.129925
3	f lag	0.07439
4	data_host_same_server_rate	0.059604
5	data_host_server_count	0.053730
6	data_host_diff _server_rate	0.046381
7	diff _server_rate	0.041244
8	Counter	0.040648
9	same_server_rate	0.036720
10	protocol_data_type	0.031750
11	data_host_same_server_port_rate	0.025766
12	Service_point	0.024004
13	pserror_rate	0.023288
14	log_in	0.021001

4.4 Result and Analysis from Experiments involving BTIntruDS

In this section, the effect of the ranking applied to the security features are analysed with respect to the efficiency of the BTIntruDS model both in time and accuracy. As pointed out earlier, the BTIntruDS model is a cyber security intrusion detection technique based on machine learning. The results of the experiments obtained from both known and unknown datasets are evaluated. The model was first built by using a 70% subset of the dataset while the rest 30% dataset was used to test the model.

In order to obtain a generalized result for the BTIntruDS model, a confusion matrix was generated which gives the ratio of incorrect positives, incorrect negatives, correct positives and correct negatives using parameters such as recall, precision, accuracy and pscore. On account of this ratio, the ability of the model to correctly predict the nature of a software intending to attack a database system can be analysed and its efficiency computed. Overall the accuracy of each set in the confusion matrix can be computed in order to determine if the prediction is correct or incorrect. The results of this

experiment is shown in table 3. From table 3, it can be observed that the precision, recall, accuracy and pscore values were given. The correct positive rate was 0.997, while the incorrect positive rate was 0.003. It can therefore be seen that the BTIntruDS model was able to measure the efficiency of the computation derived from the BTIntruDS model and the security dataset used. Also Figure 6 depicts the results obtained using an ROC curve which shows the ratio of the correct positive rate and the incorrect positive rate. From the graph in figure 6, it can be observed that the correct positive rate is almost equal to unity (0.997). From this results, it can be correctly stated that the BTIntruDS model correctly detect the nature of security dataset presented to it. It should be noted again that the nature of security dataset can either be normal or abnormal depending on the pattern in which these datasets occur. These pattern of occurrence helps a great deal to deal with yet to be identified dataset test cases, in order words the identification of an intrusion software is dependent on its occurring pattern and doesn't depend on just the past machine learning outcomes of previous datasets.

Table 3 Final result for the BTIntruDS model using he different sets

Set	Significant Value	Recall	PScore	Accuracy
Normal	0.99	0.99	0.99	0.99
Abnormal	0.99	0.99	0.99	0.99

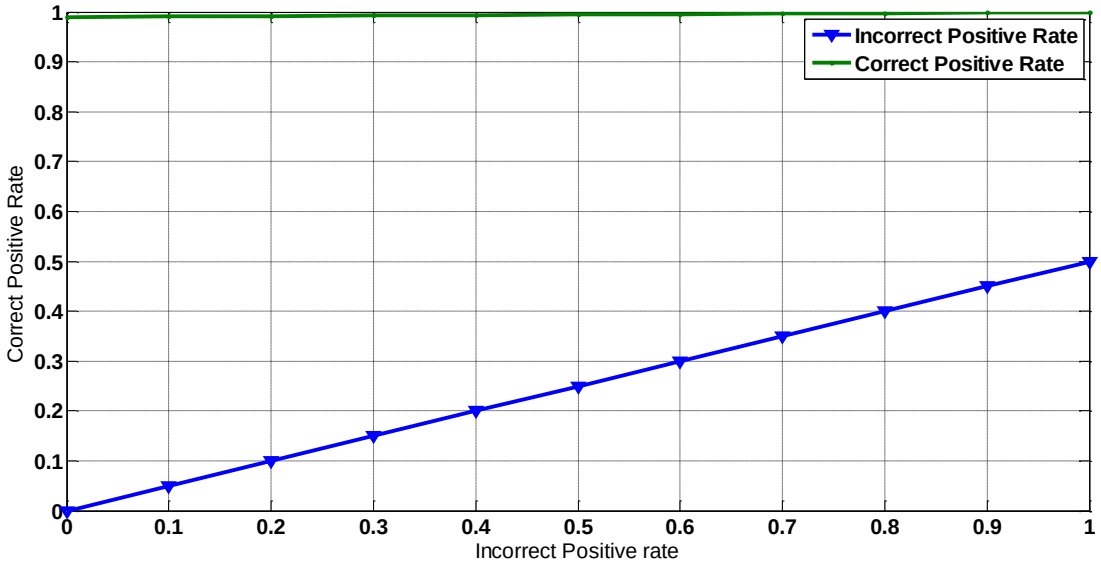

Figure 6 The ROC curve plotting correct positive rate versus Incorrect positive Rate

4.5 Comparative analysis of BTIntruDS Model

In this section, the efficiency of the BTIntruDS model is compared to five other intrusion detection model, these are NaiveBayes (NB), Logistic Regression (LR), K-Nearest Neighbour (KNN), Support Vector Machines (SVM) and the IntuDTree model. The outcome of each model was computed using the same security datasets, this was done in order to give a fair comparison among all compared models. Comparison of the models based on pscore, precision, recall and accuracy using the security datasets used earlier in the paper is shown in figure 7. The same composition of the security datasets used earlier i.e. 70% data was used to train the model while the remaining 30% was used to test the model. From figure 7, it can be observed that BTIntruDS model outperforms the IntruDTree model by 18% and the rest of the machine learning model by an average of 37%. The improved performance in comparison to the IntruDTree model was due to the use of a binary tree instead of the open tree concept used in IntruDTree. Binary tree has a better searching algorithm over other tree data structures. The use of binary tree resulted in an improved classification/ranking time for the security features. The improved performance of BTIntruDS model over the traditional machine learning model i.e. NB, LR, KNN and SVM can be attributed to the pruning of the security datasets used instead of all security datasets as used in these models. This helps reduce the variance in the importance values of the security datasets. The use of reduced datasets also helped build a generalised model in faster and more efficient manner.

The reduced dataset also helped to improve the rate of correct prediction for all classes of datasets either known or yet to be identified datasets, it also helped reduced the complexity involved in the computation of the used data-driven security model. This makes the BTIntruDS a more robust model among the other five compared protocols in terms of intrusion detection. It can therefore be observed from figure 7 and the rest of other results shown in this paper that the BTIntruDS model is more efficient and algorithmically superior to the compared models when considering security to database and other online information.

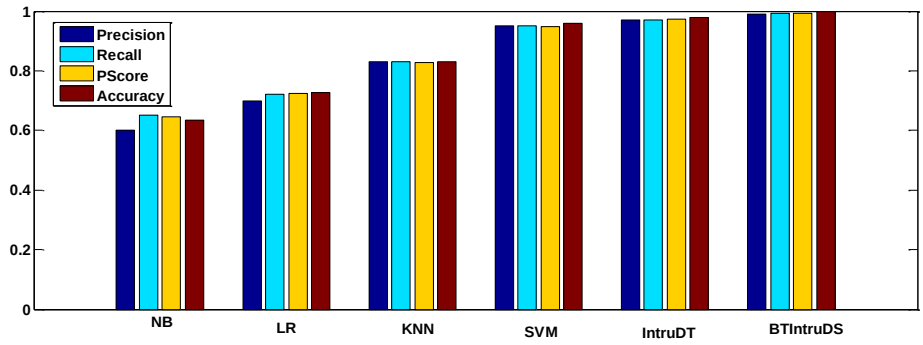

Figure 7. Comparison of the Effectiveness of BTIntruDS sing performance metrics of precision, recall, pscore, and accuracy of different machine learning based security models.

5. Conclusion

This paper highlighted the effectiveness of the BTIntruDS, a machine learning binary tree based intrusion detection model in combating adversarial intrusion into a database or online systems. The model used ranking of security features in a dataset based on their importance. This was used to prune the number of security features for final selection into the BTIntruDS model. A binary tree based on this ranking was then used to enter nodes into the tree. The nodes was built from top to bottom in accordance with the importance of the security features. The ranking of the security features was based on regularity of patterns formed so as to denote the security features as normal or abnormal. This patterns were employed in the BTIntruDS model to detect yet to be identified datasets. This helps to generalize the model for all datasets irrespective of the history of past datasets. A number of experiments were conducted to verify the effectiveness of the BTIntruDS model on cybersecurity datasets. A comparative analysis of the BTIntruDS was also done with five other models namely NB, LR, KNN, SVM and IntruDTree machine learning based intrusion detection model. In future a wider set of security datasets with higher number of security features can be looked into. This will be important in the detection of intrusion on IoT systems were varieties of devices are employed in order to determine its efficiency at such application level of cybersecurity domain.

Author Contributions: Authors contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.
Funding: Persona;
.Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES :

1. Sun, N.; Zhang, J.; Rimba, P.; Gao, S.; Zhang, L.Y. and Xiang, Y. (2010) Data-driven cybersecurity incident prediction: A survey. IEEE Commun. Surv. Tutor. Vol 21, pp1744–1772.
2. Dainotti, A.; Pescapé, A.;and Ventre, G. (2015) Worm traffic analysis and characterization. In Proceedings of the IEEE International Conference on Communications, Glasgow, UK,; pp. 1435–1442.
3. Qu, X.; Yang, L.; Guo, K.; Ma, L.; Sun, M.; Ke, M. and Li, M. A (2022) Survey on the Development of Self-Organizing Maps for Unsupervised Intrusion Detection. Mobple. Networks.
4. IBM Security Report. (2022) Available online: https://www.ibm.com/security/data-breach (accessed on 20 February 2024).
5. Tsai, C.F.; Hsu, Y.F.; Lin, C.Y.and Lin, W.Y.(2019) Intrusion detection by machine learning: A review. Expert System. Application. Vol 36, pp11994–12000.
6. Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.and Karimipour, H. (2023) Cyber intrusion detection by combined feature selection algorithm. J. Inf. Secure. Application. Vol 44, pp 80–88.
7. Tapiador, J.E.; Orfila, A.; Ribagorda, A. and Ramos, B.(2018) Key-recovery attacks on KIDS, a keyed anomaly detection system. IEEE Trans. Dependable Secur. Comput. Vol, 12, pp 312–325.
8. Tavallaee, M.; Stakhanova, N.and Ghorbani, A.A. (2017) Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. Vo 40, pp 516–524.
9. Milenkoski, A.; Vieira, M.; Kounev, S.; Avritzer, A. and Payne, B. D.(2020) Evaluating computer intrusion detection systems: A survey of common practices. ACM Computer. Survey. (CSUR) 2020, Vol 48, pp 1–41.
10. Buczak, A.L.and Guven, E.(2021) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communication. Survey. Vol 18, pp 1153–1176.
11. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H. and Wang, C.(2022) Machine learning and deep learning methods for cybersecurity. IEEE Access Vol 6, pp 35365–35381..
12. Moskovitch, R.; Elovici, Y. and Rokach, L.(2014) Detection of unknown computer worms based on behavioural classification of the host. Computer. Station. Data Analysis. Vol 52, p 4544–4566.
13. Sommer, R. and Paxson, V. (2017) Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, USA,; pp. 305–316.

14. Seufert, S. and O'Brien, D. (2014) Machine learning for automatic defence against distributed denial of service attacks. In Proceedings of the 2007 IEEE International Conference on Communications, Glasgow, UK,; pp. 1217–1222.
15. Sarker, I.H.; Kayes, A. and Watters, P. (2023) Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage. J. Big Data Vol 57, pp 23 -40.
16. Sinclair, C.; Pierce, L. and Matzner, S. (2012) An application of machine learning to network intrusion detection. In Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'12), Phoenix, AZ, USA, pp. 371–377.
17. Alazab, A.; Hobbs, M.; Abawajy, J. and Alazab, M. (2020) Using feature selection for intrusion detection system. In Proceedings of the 2020 International Symposium on Communications and Information Technologies (ISCIT), Gold Coast, Australia,; pp. 296–301.
18. Han, J.; Pei, J. and Kamber, M. (2018) Data mining: Concepts and Techniques; Elsevier: Amsterdam, The Netherlands,
19. Witten, I.H. and Frank, E. (2014) Data Mining: Practical Machine Learning Tools and Techniques; Morgan Kaufmann: Burlington, MA, USA, .
20. Agrawal, R. and Srikant, R. (2008) Fast algorithms for mining association rules. In Proceedings of the 27th International Conference on Very Large Data Bases, Santiago, Chile,; Vol 1215, pp. 487–499.
21. Sarker, I.H. and Salim, F.D. (2023) Mining User Behavioural Rules from Smartphone Data through Association Analysis. In Proceedings of the 25nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Melbourne, Australia,; pp. 450–461.
22. Sarker, I.H. (2023) Context-aware rule learning from smartphone data: Survey, challenges and future directions. J. Big Data
23. Sarker, I.H. (2022) A machine learning based robust prediction model for real-life mobile phone data. Internet Things Vol 5, pp 180–193.
24. Li, Y.; Xia, J.; Zhang, S.; Yan, J.; Ai, X. and Dai, K. (2019) An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert System Applcation Vol 39, pp 424–430.
25. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A. and Yazdani, N. (2019) Mutual information-based feature selection for intrusion detection systems. J. Network Computer Application Vol 34, pp 1184–1199.
26. Hu, W.; Liao, Y. and Vemuri, V.R. (2017) Robust Support Vector Machines for Anomaly Detection in Computer Security. In Proceedings of the International Conference on Machine Learning and Applications—ICMLA, Los Angeles, CA, USA, pp. 168–174.
27. Wagner, C.; François, J. and Engel, T. (2019) Machine learning approach for IP-flow record anomaly detection. In Proceedings of the International Conference on Research in Networking, Valencia, Spain, pp. 28–39.
28. Moskovitch, R.; Nissim, N.; Stopel, D.; Feher, C.; Englert, R. and Elovici, Y. (2016) Improving the detection of unknown computer worms activity using active learning. In Proceedings of the Annual Conference on Artificial Intelligence, Osnabrück, Germany, pp. 489–493.
29. Kotpalliwar, M.V. and Wajgi, R. (2020) Classification of Attacks Using Support Vector Machine (SVM) on KDDCUP'20 IDS Database. In Proceedings of the 2020 8th International Conference on Communication Systems and Network Technologies, Gwalior, India, pp. 987–990.
30. Kruegel, C.; Mutz, D.; Robertson, W. and Valeur, F. (2016) Bayesian event classification for intrusion detection. In Proceedings of the 24th Annual Computer Security Applications Conference, Las Vegas, NV, USA, pp. 14–23.
31. Benferhat, S.; Kenaza, T. and Mokhtari, A. (2013) A naive bayes approach for detecting coordinated attacks. In Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, Turku, Finland, pp. 704–709.
32. Panda, M. and Patra, M.R. (2017) Network intrusion detection using naive bayes. International Journal of Compututer Science Network Vol 7, pp 258–263.
33. Koc, L.; Mazzuchi, T.A. and Sarkani, S. (2020) A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert System Application Vol 39, pp 13492–13500.
34. Vishwakarma, S.; Sharma, V. and Tiwari, A. (2021) An intrusion detection system using KNN-ACO algorithm. International Journal of Computer Application Vol 171, pp 18–23.
35. Bapat, R.; Mandya, A.; Liu, X.; Abraham, B.; Brown, D.E.; Kang, H. and Veeraraghavan, M. (2022) Identifying malicious botnet traffic using logistic regression. In Proceedings of the 2022 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, pp. 266–271.
36. Besharati, E.; Naderan, M. and Namjoo, E. (2023) LR-HIDS: Logistic regression host-based intrusion detection system for cloud environments. Journal of Ambient Intelligence. Vol 10, pp 3669–3692.
37. Kumar, P.A.R. and Selvakumar, S. (2018) Distributed denial of service attack detection using an ensemble of neural classifier. Computer Communication Vol 34, pp 1328–1341.
38. Dainotti, A.; Pescapé, A. and Ventre, G. (2017) A cascade architecture for DoS attacks detection based on the wavelet transform. Journal of Computer. Security Vol 17, pp 945–968.
39. Quinlan, J.R. (1998) Induction of decision trees. Machine Learning Vol 1, pp 81–106.
40. Quinlan, J.R. (2009) C4.5: Programs for Machine Learning; Morgan Kaufmann Publishers, Inc.: Burlington, MA, USA,
41. Sarker, I.H.; Colman, A.; Han, J.; Khan, A.I.; Abushark, Y.B. and Salah, K. (2022) BehavDT: A Behavioural Decision Tree Learning to Build User-Centric Context-Aware Predictive Model. Mobile Network
42. Ingre, B.; Yadav, A. and Soni, A.K. (2021) Decision tree based intrusion detection system for NSL-KDD dataset. In Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems, Ahmedabad, India, pp. 207–218.
43. Iqbal H. S.; Yoosef B. A.; Fawaz A. and Asif I. K (2023) IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model in symmetry MDPI, Vol 2 pp 1 – 15.