Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

A Survey Paper on Machine Learning Approaches to Intrusion Detection

Oyeyemi Osho
Computational Data and Enabled Science & Engineering
Jackson State University
Jackson, Mississippi, USA

Sungbum Hong (PhD)
Computational Data and Enabled Science & Engineering
Jackson State University
Jackson, Mississippi, USA

***Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. For any nation, government, or cities to compete favorably in today's world, it must operate smart cities and e-government. As trendy as it may seem, it comes with its challenges, which is cyber-attacks. A lot of data is generated due to the communication of technologies involved and lots of data are produced from this interaction. Initial attacks aimed at cyber city were for destruction, this has changed dramatically into revenue generation and incentives. Cyber-attacks have become lucrative for criminals to attack financial institutions and cart away with billions of dollars, led to identity theft and many more cyber terror crimes. This puts an onus on government agencies to forestall the impact or this may eventually ground the economy.**

The dependence on cyber networked systems is impending and this has brought a rise in cyber threats, cyber criminals have become more inventive in their approach. This proposed dissertation discusses various security attacks classification and intrusion detection tools which can detect intrusion patterns and then forestall a break-in, thereby protecting the system from cyber criminals.

This research seeks to discuss some Intrusion Detection Approaches to resolve challenges faced by cyber security and e-governments; it proffers some intrusion detection solutions to create cyber peace. It discusses how to leverage on big data analytics to curb security challenges emanating from internet of things. This survey paper discusses machine learning approaches to efficient intrusion detection model using big data analytic technology to enhance computer cyber security systems.

Keywords—Component; Intrusion Detection; cyber security; machine kearning; Cyber attacks; Security.

I. INTRODUCTION

The effects of cyber-attacks are felt around the world in different sectors of the economy not just a plot against government agencies. According to McAfee and Center for Strategic and International Studies (2014), nearly one percent of global GDP is lost to cybercrime each year. The world economy suffered 445 billion dollars in losses from cyber-attacks in 2014. Adversaries in the cyber realm include spies from nation-states who seek our secrets and intellectual property; organized criminals want to steal our identities and money; terrorists who aspire to attack our power grid, water supply, or other infrastructure; and hacktivist groups who are trying to make a political or social statement (Deloitte 2014).
According to Dave Evans (2011), Explosive growth of smartphones and tablet PCs brought the number of devices connected to the internet to 12.5 billion in 2010, while the world's human population increased to 6.8 billion, making the number of connected devices per person more than 1 (1.84 to be exact) for the first time in history. Reports show that the number of internets connected devices will be 31 billion worldwide by 2020.

Internet and web technologies have advanced over the years and the constant interaction of these devices has led to the generation of big data. Using big data according to John Walker (2014) leads to better decisions. Using big data makes room for better decisions, the current technology generates huge amounts of data which enables us to analyze the data from different angles.

Due to the amount of information put out by technologies, security of data has become a major concern. New security concerns are emerging, and cyber-attacks never cease, according to Wing Man Wynne Lam (2016) "it is common to see software providers releasing vulnerable alpha versions of their products before the more secure beta versions". Vulnerability refers to the loopholes in systems created, all technologies have their weak points which may not be openly known to the user until it is exploited by hackers.

Cyber security concerns affect all facets of the society including retail, financial organizations, transportation industry and communication. H. Teymourlouei et al. [32], better actionable security information reduces the critical time from detection to remediation, enabling cyber specialists to predict and prevent the attack without delays. The rate of increase in devices which requires internet connection has led to the emergence of internet of things. This makes the world truly global and in one space, although internet of things has provided many opportunities like new jobs, better revenue for government and people involved in the industry, reduced cost of doing business, increased efficiency handling the big data associated with this trend has become the issue.

Almost all internet of things applications has sensors which monitors discrete events and mining data generated from transactions. The data generated through this device can be used in investigative research which will eventually impact decision making on the part of the industries concerned.
Vulnerability markets is a huge one because some software developers sell their vulnerability for hackers in some cases, hence the hacker's prey on users of the software. Hackers used to be destructive in their approach, has we have seen in recent times has been purely for making money. Some ask you to call them so that they can offer you support at certain fee bargained. Sometimes hackers access government systems through the network and seize important information's stored on the system hence demand for ransom. Other times could be detecting bugs in software's purchased by government

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

agencies and demand for ransom else they release the error to the public which may lead to a huge loss in data and money.

The emergence of technologies has led to smart cities, which simply implies to the application of electronic data collection to supply required information used to manage available resources effectively. The concept of smart cities is what has been adopted by many states and nations, web-based government services brings about efficient run of government. This is something evident in first world nations of the world. To be highly competitive in today's world, no reasonable government will shy away from e-governance. As beautiful as this may sound, there are challenges militating against it, one prominent problem is hacking and e-terrorism. Due to information's been put out by users, information that include tax information, social security numbers and other personal information on the web, this creates caution from government end to secure the information being posted by citizens on government websites.

Cybersecurity of government facilities including software's, websites and networks is very challenging and most cases very expensive to maintain. That is why this paper proposes big data analytic tools and techniques as a solution to cyber security.

According to Tyler Moore (2010)," Economics puts the challenges facing cybersecurity into perspective better than a purely technical approach does. Systems often fail because the organizations that defend them do not bear the full costs of failure. Many of the problems faced by cybersecurity are economic in nature and solutions can be proffered economically. In this paper, I will offer a big data analytics perspective and recommendations that can help to ameliorate the state of internet of things and cybersecurity. Looking at the business side of cybersecurity, I think it is either not properly funded or underfunded, if cooperation's and government agencies pump enough funding into cyber city, they will become better for it and this might reduce cyber-attacks.

According to Wamba et al. (2017), Elgendy & Elragal(2014), Holsapple et al.(2014), big data analytics is a holistic approach and system to manage, process and analyze huge amount of data in order to create value by providing a useful information from hidden patterns to measuring performance and increase competitive advantages.

Some techniques mentioned in this paper include biometric authentication, data privacy and integrity policy, Apache Storm algorithm, continuous monitoring surveillance, log monitoring, data compression and event viewer. In this paper I will be implementing big data analytics using R programming and Python programming, gephi, tableau, rapid miner for analysis and data visualization.

II. INTRUSION DETECTION WITH GDA-SVM APPROACH

Some earlier research on cyber security Intrusion detection through machine learning analytical tools are described below. **Zulaiha** et al. [13] used the Great Deluge algorithm (GDA) to implement feature selection and Support Vector Machine for its classification. Great Deluge algorithm (GDA) was proposed by Dueck in 1993, it is a generic algorithm applied to optimization problems. It has similarities to the high-climbing and simulated annealing algorithms, the main difference between the Great Deluge algorithms and the Simulated Annealing algorithms is the deterministic acceptance function of the neighboring solution.

The inspiration of Great Deluge algorithm (GDA) is derived from the inspiration of a person climbing up a hill preventing his feet from getting wet as the water level rises. Finding the optimum of an optimization problem is seen as finding the highest point in a landscape. The GDA accepts the 'level' which is where the absolute values of cost function is equal or less than the initial objective function, the initial objective function is equal to the initial value of the level. The advantage of Great Deluge algorithm (GDA) is that it only depends on the 'up' value which represents the speed of the rain, if the 'up' is high the algorithm will be fast with poor results but if the 'up' value is small the algorithm will produce better results with good computational time.

Zulaiha Et al. [18] used SVM (support vector machine) classifier, the fitness of every feature is measured by means of 10-fold cross validation, the 10-fold cross validation is used to generate the accuracy of classification by SVM. In the 10FCV which contains 10 subsets, one is used for testing while the remaining is used for training, the accuracy rate is computed over 10 trials. Zainal et al. [21] describes complete fitness function as:

$$\alpha * \gamma R(D) + \beta * \frac{|C| - |R|}{|C|}$$

Where γR(D) is the average of accuracy rate obtained by conducting ten multiple cross-validation with SVM, D is the decision, |R| is the '1' number of position or the length of selected feature subset, |C| is the total number of features, α and β are two parameters corresponding to the importance of classification quality and subset length $\alpha \in [0,1]$
and $\beta = (1-\alpha)$, respectively.

Zulaiha Et al. [18] proposed GDA-SVM Feature Selection Approach:

Step 1: (Initialization) randomly generates an initial solution, all features are represented by binary string, where '1' is assigned to a feature if it will be kept and '0' is assigned to a feature which will be discarded, while N is the original number of features.

Step 2: Measure the fitness of the initial solution, where the accuracy of the SVM classification and all the chosen features are utilized to calculate the fitness function.

Step 3: A random solution is generated when the algorithm search about the initial solution by mutation operator.

Step 4: Evaluate the fitness of the new solution and accept the solution where the fitness is equal or more than the level. Update the best solution if the fitness of the new solution is higher than the current best solution and level with a fix increase rate.

Step 5: Repeat these steps until a stopping criterion is met. If stopping condition is satisfied, the solution with best fit is chosen; otherwise, the algorithm will generate new solution.

Step 6: Train SVM based on the best feature subset, after this, conduct testing data sets.

Pseudo code of GDA-SVM approach for feature selection.

1. Initialize a random solution
2. Evaluate the fitness of the solution

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

3. While (stopping criteria not met)
4. Generate at random a new solution about initial solution.
5. Evaluate the fitness of the new solution. Accept the solutions where fitness is equal or more than level.
6. End while
7. Output best feature subset.

Zulaiha Et al. [18] used the KDD-CUP 99 data subset that was pre-processed by the Columbia University and distributed as part of the UCI KDD Archive. The training data contains about 5 million connection records and 10% of the training data has 494,012 connection records.

Chebrolu et al. [20] assigned a label (A to AO) to each feature for easy referencing, the training data has 24 attack types with four main categories: 1) DOS: Denial of service, 2) R2L: Unauthorized access from a remote machine (remote to local), 3) U2R: unauthorized access to local privileges (user to root), 4) Probing: surveillance. Features used by Almori & Othman, 2011 was implemented and trained based on the dataset. The intrusion detection could either be an attack or normal.

GDA Performance based on Highest Fitness Function: Experiment conducted by Almori & Othman based on Highest Fitness was used to train the SVM classifier. The features involved in the training process are B, G, H, J, N, S, W, G and L.

According to Zulaiha Et al. [18], the table below compares the classification performance for the seven feature subsets produced by previous techniques. The mean gives the average performance of the feature subset proposed by the respective technique on three different test sets.

Comparison of Classification Rate.

TABLE I. COMPARISON OF CLASSIFICATION RATE [22]

Technique	Data2	Data3	Data4	Mean
LGP (C, E, L, AA, AE&AI)	75.37	94.60	87.65	65.87
SVDF (B, D, E, W, X& AG)	82.60	89.83	84.98	85.80
MARS (E, X, AA, AG, AH&AI)	70.47	92.65	90.87	84.66
Rough Set (D, E, W, X, AI &AJ)	81.65	89.56	87.18	86.13
Rough-DSPO (B, D, X, AA, AH & AI)	71.85	93.232	92.07	85.72
BA (C, LX, Y, AF & AK)	85.25	96.65	98.20	93.36
GDA (B, G, HJ, N, S, W, GL)	83.16	90.75	87.45	87.12

Pratik et al. [22] proposed the first three rows subset, in the dataset the mean value depict that Great Deluge algorithm (GDA) has the second highest average classification rate. This show that GDA performs better than other techniques aside BA.

SVM's are based on the idea of structural risk minimization which results in minimal generalization error [44], the number of parameters does not depend in input features rather on margin between data points. Hence, SVM's does not require reduction of feature size to avoid overfitting, they provide a system to fit the surface of the hyperplane to the data using the Kernel function. The advantages of SVM is the binary classification and regression which results in low expected probability of generalization errors. SVM possess real time speed performance and scalability, they are insensitive to number of data points and dimension of the data.

Support vector machine (SVM) approach is a classification technique based on Statistical Learning Theory (SLT). It is based on the idea of a hyper plane classifier, or linearly separability. The goal of SVM is to find a linear optimal hyper plane so that the margin of separation between the two classes is maximized [45]. Suppose we have training data points$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)\}$,where $x_i \in R^d$ and $y_i \in \{+1, -1\}$. Consider a hyper plane defined by (w, b), where w is a weight vector and b are a bias. A new object x can be classified with the following function:

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right)$$

Where $K(x_i, x)$ is the kernel function.

III. SEQUENTIAL PATTERN MINING APPROACH

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Wenke Lee et al. [49] proposes that to have an effective base classifier, enough data must be trained to identify meaningful features. Here are some algorithms that can be used to mine pattern from Big Data":

- Association rules: The goal of the mining association rule is to determine feature correlations from a Bigdata. An association rule is an expression of $X \Rightarrow Y, confidence, support$[49]. X and Y are subsets of items in a record, $support$ is the percentage of records that contain $X + Y$ while $confidence$ is $\frac{support(X+Y)}{support\ (X)}$.
- Frequent Episodes: A frequent episode is a set events that occur frequently withing a time frame. Events in serial episode must occur in partial order in time while events parallel episode does not have such constraint. For X and Y, $X + Y$ is a frequent episode, of $X \Rightarrow Y$ with $confidence$ is $\frac{support(X+Y)}{support\ (X)}$ and $support = frequency(X + Y)$ is a frequent episode rule[49].
- Using the discovered patterns: The association rule and frequent rules can be combined into one unit using the merge process:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

- ✓ Find a match in the aggregate rule set where a match is when both LHS and RHS rules matches, ε matches on the $support$ and $confidence$ values.

Shi-Jie Song et al. [47] proposed a misuse intrusion detection model based on sequential pattern mining using two steps, the first step is to search for one large sequence:

- Each item in the original database is a candidate of one-large-sequence-one-itemset C_1, a C_1. Is taken out. The items in the database is scanned vertically, if any instance contains C_1, the support of C_1. Adds 1. If the support is greater than the given minimal support, the C_1 is one-large-sequence-one-itemset L_1, tis process is repeated and binary records of L_1's is stored in a temporary database.
- Two one-large-sequence-$(k-1)$-itemset L_{k-1} is combined into one-large-sequence-k-itemset C_k. In the C_k, the order of the $k-1$ bits are arranged as the original order.
- When C_k is taken out each L_{k-1} is scanned vertically in the temporary database and horizontally for number of k bits. If C_k exist in a instance, the support of C_k adds 1, if the support is greater than the minimal support then C_k is L_k. The L_k is found in turn and listed in one-large-sequence-itemset.

The second step is to search for k-large sequence SL_k:

- The two k-1 sequence SL_k in the temporary database is combined to form the candidate k-large-sequence SC_k. Sort out two SL_k's with the same former k-2 bits, combine the bits to form two k-1 bits, the k-1 bits can be arranged in four orders.
- When the SC_k is taken out, each SL_{k-1} is searched vertically in the temporary database, if k bit exist in any instance, the support of SC_k adds 1. If the support is greater than the minimal support, SC_k is SL_k, the process is repeated to find SL_k.

Fidalcastro et al. [46] proposed using Fuzzy logic with sequential data mining in Intrusion Detection Systems. In this paper Fuzzy logic is used as a filter for feature selection to avoid over fitting of pattern and reduce the dimension complexity of the data. The filter-based approach of Fuzzy logic is used for feature selection from the training data, some filtering criteria applied are:

Information Gain it measures the expected reduction in entropy of class before and after observing features. It selects features by larger difference; it is measured as [46]

$$\boldsymbol{InfoGain(S,F) = Entropy(s) - \sum_{v\in V}\frac{|Sv|}{|S|}Entropy(Sv)}$$

Where S is the pattern set, Sv is the subset of S, F as a value v, |S| is the number of samples in S, v is the value of the feature F. The entropy of class before observing features is defined as:

$$Entropy(S) = \sum_{c+C}^{n} -\frac{|Sc|}{S}\log_2\frac{|Sc|}{|S|}$$

Where Sc is subset of S belonging to class c, C is the class set and IG is the fastest and simplest ranking method [46].

Gain ratio (GR) normalizes the IG by dividing it by the entropy of S with respect to feature F, gain ratio is used to discourage the selection of features uniformly distributed values, it is defined as:

$$GainRatio(S,F) = InfoGain(S,F)/SplitInfo(S,F)$$

$$SplitInfo(S,F) = \sum_{i=1}^{n} -\frac{|Si|}{|S|}\log_2\frac{|Si|}{|S|}$$

Where Si is the subset of S where feature F has its i^{th} possible value, n is the number of subclasses split by feature F.

Chi-Square (CS): measures the chi square value of each feature with respect to the classes, the values are ranked and the strong correlation with the classes are the large chi square values of the features. The chi square of feature F is defined as:

$$ChiSquare(f) = \sum_{i=1}^{m}\sum_{j=1}^{k}(A_{ij} - E_{ij})/E_{ij}$$

$$E_{ij} = (R_i, C_j)/|S|$$

K is the number of classes, C_j is the number of samples in the j^{th} class, m is the number of intervals discretized from the numerical values of F, R_i is the number of samples in the i^{th} interval, A_{ij} is the number of samples in the i^{th} interval with j^{th} class and E_{ij} is the expected occurrence of A_{ij} [46].

IV. CLUSTERING APPROACH

The objective of Cluster analysis is to find groups in data [53], the groups are based on similar characteristics. For a dataset D featured by P attributes:

$$D_n = [A_1, A_2, \dots, A_p]$$

D is partitioned into $\{C_{1,j=1-k}\}$ clusters so that:

For each $D_j = [A_1, A_2, \dots, A_p]$

$$SM_{D_j/C_i}(A_1, \dots, A_p) = \max\{\frac{SM_{D_j/C_i}(A_1, \dots, A_p)}{i \neq l}\}$$

Where SM is the similarity between D_i and C_k, and $\{C_i\}$ should meet the following conditions:

$$C_i \neq \emptyset;\ C_i \cap C_i = \emptyset; \cup_{i=1}^{k} C_i = D\ ,\qquad i, l = j \dots . k$$

Clustering has three main partitioning approaches:

Hierarchical approach where hierarchy of clusters are built, and each observation starts in its own cluster and pairs are merges and moved up the hierarchy.

Non-hierarchy approach requires a random initialization of the clusters and set of rules to define the criterion.

Biomimetic approach is inspired from the ethology like ant colonies, the models developed from ideals provide better solutions to problems in Artificial Intelligence [53].

Clustering analysis is a pattern recognition procedure whose goal is to find patterns in a dataset. It identifies clusters and builds a typology [54] of sets using a certain set of data, a cluster is a collection of data objects that are like one another. A good clustering method produces high quality cluster to ensure that the inter-cluster similarity is low, and the intra-cluster similarity is high, this infers that members of a cluster are more like each other than they are with different clusters [54].

Fixed-Width Clustering procedure:

The Fixed-Width clustering algorithm is based on a set of network connections [56] C_T for training, each connection c_i is represented by a di-dimensional vector feature. The Fixed-Width clustering involves three stages:

Normalization: This ensures features have the same influence when calculating distance between connections. Each continuous feature x_j is normalized in terms of the number of standard deviations from the mean of the feature.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

Cluster Formation: This process takes place after normalization, the distance between each connection c_i is measured in the training set C_T to the center of each cluster centroid. If the distance [56] to the closest cluster is less than the threshold w, then the centroid of the closest cluster is updated and the total number of points in the cluster is incremented, otherwise a new cluster is formed.

Cluster Labelling: This is the process of labelling network clusters based on its value, if a cluster contains more than the classification threshold fraction τ of the total points in the data set, such cluster is labelled normal else it is labelled anomalous.

Test Phase: This is the final stage of the fixed-width clustering approach where each new connection is compared to each cluster to determine if it is normal or anomalous. If the calculated distance from the connection to each cluster is less than the cluster width parameter w , then such connection shares the label of its closest cluster, otherwise the connection is labeled anomalous.

S.Sathya et al. [56] proposed a clustering method in which the frequency of common pairs of each cluster is found using cluster index and choosing a cluster having maximum number of common pairs with most $k-nearest$ neighbors for merging. This process reduces the number of computations considerably and performs better.

Nong Ye et al. [57] proposed a scalable Clustering technique titles CCA-S (Clustering and Classification Algorithm-Supervised). The CCA-S has a dataset considered as datapoints in a dimensional space, for Intrusion Detection, the target variable is a binary with two possible values: 0 for normal and 1 for intrusion. CCA-S clusters data based on two parameters: the distance between data points and the class label for data points. Each cluster represents a pattern for normal or intrusion activities depending on the class label of the data points in the cluster. The three stages of the CCA-S are as follows:

Training (Supervised Clustering): It takes two steps to incrementally group the N data points in the training data set into set into clusters.

- The first stage calculates the correlation between the predictor variable X_i and the target variable Y_i. Dummy clusters are formed with one being for the normal activities and the other dummy vector for intrusive activities, the centroid of the clusters is determined by the mean vector of all activities in the training dataset for both clusters.
- Incrementally group each training data points into clusters, given a data point X, the nearest cluster L to this data point is determined by using a distance metric weighted by the correlation coefficient of each dimension. If L is same class as X, then X is grouped will L, else we create a new cluster with this data point as the centroid of the training dataset.

Classification: There are two methods to classify a data point X in a testing dataset [57].

- Assign the data point X the class dominant in the k nearest clusters which are found using a distance metric weighted by the correlation coefficient of each dimension.
- Use the weighted sum of the distances of k-nearest clusters to this data point to calculate a continuous value for the target variable in the range of [0,1].

Incremental Update: At this stage, the correlation and clustering of datapoints are calculated and the result is stored, as new training data are presented, each step of the training for the new data points is observed and the clusters are updated for new data points incrementally.

V. HADDOP APPROACH TO INTRUSION DETECTION

Apache Hadoop is an open-source software framework, it processes big data and manages programs on a distributed system. It has two components, MapReduce, and Hadoop Distributed file system.

M. Mazhar et al. [57] proposed a Hadoop system to process network traffic at real-time for intrusion detection with higher accuracy in the high-speed Big Data environment. The traffic is captured with high-speed capturing device, the captured traffic is sent to the next layer filtration and loads balancing server (FLBS). Only the undetermined traffic is filtered by efficient searching and comparisons in In-Memory intruder's database. The unidentified network flow and packet header information to the third layer (Hadoop layer) master servers. The role of the FLBS is to decide the packets to be sent to master server depending on the IP addresses thereby causing a load balance. When the network traffic gets to the master, it generates sequence file for each flow so that it can be processed by the Hadoop data nodes, at the Hadoop node the packet is extracted to secure the information carried be the packet using Pcap (Packet capture) [57]. This process continues over a period of network flow, the process will result in set of sequence file which are lined up in parallel, the processed sequence files are then analyzed by Apache Spark which is a 3rd party tool in Hadoop system. The feature values are sent to layer four which is the Decision servers which then classifies the packets into normal or intrusion based their parameter values, the decisions are then stored in In-memory Intrusion database.

Sanraj et al. [58] proposed a system based on integration of two different technologies, Hadoop and CUDA that work together to boost Network Intrusion Detection System. The system is separated into different modules in which Hadoop will take care of data organization and GPGPU (General purpose graphic processing unit) takes care of intrusion analytics and identification. The network packet and data logs gain access to the Hadoop system using Flume. The function of Flume in the Hadoop system is to provide the real time streaming the service of data collection and routing. The traffic ingestion is done over HDFS (Hadoop distributed file system), the system pre-organization is done on server logs and packet data. Data compilation done on the HDFS is moved forward to GPGPU for network intrusion detection. The NIDS (Network intrusion Detection System) was designed with five phases which is explained below:

1. Traffic Ingestion: This stage refers to how packet data is streamed from the various onsite servers to the Hadoop cluster via the DFS (Distributed file system). To make this transfer possible, a flume agent is used

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

as the interface between the individual onsite servers and the Hadoop framework.

2. Packet Analysis: The Sanraj et al. [58] NIDS utilizes the Python-Scapy for packet classification, the NIDS captures packets available on the Network Interface card (NIC). The packet data is collected over various network traffic and the packet classification extracts information from the packet data about the protocol number, payload, source, destination, and hardware address.
3. Data Compilation: This is the stage where the NIDS compiles data, logs and network information which was received from the various servers into datasets on the Distributed file system (DFS). The packet data collected with the fume agents are logged into local hard discs and complied into dataset.
4. Intrusion Analysis: The compiled dataset is forwarded to the General-Purpose Graphical Processing unit (GPGPU) to detect intrusion pattern on the dataset received and submit the result back to the Hadoop. The signature mapping is performed by Compute Unified Device Architecture (CUDA technology), the result computed contains detected pattern, count and type of intrusion detected in the dataset.
5. Data Analytic: This is the final phase of the Network intrusion Detection System (NIDS) where result for the Hadoop system is dump back into the Distributed File System, the result contains the intrusion pattern, count, and network address. These analytics help the security administrator to configure the network security infrastructure to restrain the anomaly activity on the network [58].

Sanjai et al. [59] proposed the Real-time Network Intrusion Detection Using Hadoop-Based Bayesian Classifier, the system was designed with three phases. The KDD '99 intrusion Detection Dataset was used to evaluate the system, Apache Hive was used to process the training data, the HiveQL is used replace missing values with default values and to remove duplicate values from the dataset. A MapReduce job is used to implement a Naïve Bayes learner that uses prerecorded network traffic. In the first phase of implementing the NIDS the performance of heterogeneous cluster was compared to homogeneous cluster, it is inferred that homogenous cluster outperformed heterogenous cluster. The second phase the Hadoop-based Naïve Bayes classifier was ran on the homogeneous cluster to classify the data based on the set rules of the classifier to check for intrusion or determine normal traffic. The third stage is where the training speed of the Hadoop-based classifier and stand-alone non-Hadoop-based Naïve Bayes are compared. The Hadoop-based Naïve Bayes algorithm performed faster than the stand-alone Naive Bayes, the system also shows that Hadoop-based Naïve Bayes is not as fast as an adaptive and self-adaptive Bayesian algorithm. The fourth stage is where a network sniffer like Snort is used to capture network packets, Naïve Bayes classifier was used to capture packets in real time while tshark converts the binary data generated into a CSV file. The fifth stage is where the detection rate of the Hadoop-Based Naïve Bayes Classifier and the generated dataset is classified into normal to anomaly pattern. This information is useful for the network administrator to know if there is an intrusion and carve a defense mechanism against such attack.

VI. DECISION TREES APPROACH

Decision Tree is one of the classification algorithms in data mining, the main approach is to select the attributes which best divides the data into their classes. The process of Decision Tree is recursively applied to each partitioned subset of the data items [60]. The process stops when all the data items in current subset belongs to the same class. Each node has a few edges which are labeled according to a possible value of the attribute in the parent mode. An edge connects nodes together and a leaf, leaves are labeled with a decision value to categorize the data. Decision Trees utilizes some parameters for classification, Entropy measures the impurity of data items. The Entropy is higher when the data items have more classes. Information gain measures the ability of each attribute in classifying data items, it measures the entropy of the attributes compared with impurity of the complete dataset. The attributes with the largest information gain are considered the most useful in classifying data items. Classification of an unknown object, one starts at the root of the decision tree and follow the branch indicated by the outcome of each test until a leaf node is reached [60].

Shilpashree et al. [61] proposed the Classification and Regression Trees (CRT) approach to decision tree, the formula below is required for this approach:

Let the dataset be $S = \{(a1, b2), (a2, b2), \ . \ . \ ., (a_N, b_N),$

Where $a_i = (a_i^{(1)}, a_i^{(2)}, . \ . \ ., a_1^{(n)})^\tau, i = 1,2, . \ . \ ., N,$

a_i is the instance of the input and indicates a record for network packet, there are n features in a_i. The variable N is the amount of packet records included in the S dataset. $b_i \in \{0,1,2, \ . \ . \ . \ M-1\}$ is the output of every detected record.

Two parameters are used for intrusion detection which includes *score indicator* and the *Computation time*. The Computation time t gives the build time and time taken for detection, there are four instances classification, positive True, Positive false, Negative true and Negative false. The score indicator has four instances: Positive True describes case that is distinguished accurately, Positive False describes unusual case mistakenly classified as ordinary, Negative False describes ordinary case misclassified as unusual one while Negative True is unusual case which is distinguished accurately [61]. The score indicator is computed as thus:

Exactness E is the extent of applicable occurrence among detected samples and is defines as:

$$E = \frac{PT}{PT + PF}$$

T indicates the extent of significant occurrence over the relevant samples.

$$T = \frac{PT}{PT + NF}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

The score indicator is computed by taking average of E and T using the formula below:

Score Indicator $= \frac{(\theta^2+1)E*T}{\theta^2(E*T)}$

Manish et al. [62] proposed that Decision Trees can be constructed from large volume of dataset with many attributes this is as result of the tree size being independent from the data size. The amount of information associated with an attribute value is related to the probability of occurrence. Manish et al. proposed the ID3 algorithm for Intrusion Detection which builds decision tree by implementing information theory, entropy is a concept used to measure the amount of randomness from a dataset. The objective of decision tree classification is to iteratively partition the given data into subsets where all elements in each final subset belong to the same class. The entropy calculation is as follows:

$$Entropy: H(p1, p2, \ldots, ps) = \sum_{i=1}^{s} (p_i \log (\frac{l}{p_i}))$$

Where $p1, p2, \ldots, ps$ represent the probabilities of the different classes. The information gain of the system is given as:

$$Gain(D, S) = H(D) - \sum_{i=1}^{s} p(D_i)H(D_i)$$

The gain ratio is used for splitting purpose, the upgraded version of the ID3 is the C4.5 [62] uses highest Gain ratio for splitting purpose which ensures a more robust information gain and is given as:

$$GainRatio(D, S) = \frac{Gain(D, S)}{H(\frac{|D_1|}{D}, \ldots, \frac{|D_s|}{D})}$$

Christopher et al. [63] proposed an Improved signature-based intrusion detection using Decision Trees, the IDS attempts to find the matching rules for an input data item and the process of intrusion Detection begins at the root of the tree. A child node is selected based on the rule set and the label of the node determines the next feature that needs to be tested. This iterative process of evaluating the features is done for all specified potential matching rules of the Decision Tree beginning from the root node. When a data packet is sent from a source node to destination node, the packet is evaluated as input element, the process of intrusion detection eventually terminates at the leaf node. The Christopher et al. [63] method solves the problem of a packet data ending up in the wrong address destination by expanding the tree for all defined features that has not been used and checking the destination port to ensure that the exact information on the packet matches the destination port. The Intrusion Detection process stops when it cannot detect any successor node with a specification that matches the input elements that is being considered.

Kajal et al. [64] proposed the Decision Tree Split which is based on C4.5 decision tree algorithm with steps as follows:

1. If all the given training examples belong to the same class, then a leaf node is created for the decision tree by choosing that class.
2. For every feature 'a', calculate the gain ratio by dividing the information gain of an attribute with splitting value of the attribute. Gain ratio is given as
$$GainRatio - \frac{IG(a)}{Split(a)}$$
Where S is the set of all the examples in the given training set.
3. Information gain of an attribute is computed as
$$IG(a) = Ent(S) - \sum_{a_{val} \in values(a)}^{num_{class}} \frac{|S\_a|}{a} * Ent(S\_a)$$
Where S_a is the subset of S, values (a) is the set of all possible values of attribute 'a' and |a| is the total number of values in attribute 'a'.
4. Entropy can be calculated as
$$Ent(S) = \sum_{j=1}^{num\_class} \frac{freq(Lj, S)}{|S|} * log2(\frac{freq(Lj, S)}{|S|})$$
Where $L = L_1, L_2, \ldots, L_n$ is the set of classes and num_class is the number of distinct classes, the num_classes has only two values, normal and anomaly.
5. Split value of an attribute is chosen by taking the average of all the values in the domain at that attribute.
$$Split(a) = \frac{[\sum_{i=1}^{m}(a_{val})i]}{m}$$
Where m is the number of values of an attribute 'a'.
6. Find the attribute with the highest gain ratio, assume the highest gain ratio is for the attribute 'a_best'.
7. Construct a decision node that divides the dataset on the attribute 'a_best'.
8. Repeat steps from 1 to 4 on each subset produced by dividing the set on attribute 'a_best' and insert those nodes as descendant of parent node.

The split value of an attribute is calculated as

$$Split(a) = -\sum_{a\_val \in values(a)}^{num\_class} \frac{|S\_a|}{|a|} * log2\left[\frac{|S\_a|}{|a|}\right]$$

VII. OPEN CHALLENGES

The challenge faced by machine learning approach to Intrusion Detection is the reactive approach. Existing attack patterns are used to train the model, hence there is need to update the Intrusion Detection System to combat a new signature pattern of an attack. The Anomaly based Intrusion Detection System through the machine learning created model identifies intrusion by any deviation from the established regular pattern, this process helps in identifying intrusions in its slightest form and can also increase the rate of false positive because of classifying normal activities as intrusion.

VIII. CONCLUSION

In this article, I was able to present intrusion detection techniques and underline their properties, characteristics, and mode of operation. I also discussed the future and challenges related to Network Intrusion Detection Systems.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

REFERENCES

[1] C.L Phillip Chen, Chun-Yang Zhang
[2] K. Krishnan Data-intensive applications; challenges, techniques, and technologies: A survey on Big Data, 2014. Data Warehousing in the age of Big Data, 2013.
[3] Suresh Lakavath, Ramlal Naik.L. A Big Data Hadoop Architecture for Online Analysis, 2015.
[4] E. Goldin, D.Feldman, G.Georgoulas, M.Castana and G.Nikakopoulas Cloud Computing for Big Data Analytics in the process Control Industry, 2017.
[5] L. Breiman "Random forests", Mach. Learn., vol. 45, no. 1, pp, 5-32, 2001.
[6] Srinivas Mukkamala, Andrew Sung and Ajith Abraham Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools, 2005.
[7] Shaik Akbar, K. Nageswara Rao, J.A. Chandulal Intrusion Detection System Methodologies Based on Data Analysis, 2010
[8] Ajith Abraham, Crina Grosan. Yuechui Chen Cyber Security and the Evolution of Intrusion Detection Systems, 2005
[9] Julien Corsini Analysis and Evaluation of Network Intrusion Detection Methods to Uncover Data Theft, 2009
[10] G.Nikhitta Reddy, G.J.Ugander Reddy A study of Cyber Security Challenges and its Emerging Trends on Latest Technology, 2014
[11] G. McGraw and G. Morrisett Attacking malicious code: A report to the infosec research council. IEEE Software, 17(5):33-44, 2000.
[12] Kutub Thakur, Meikang Qui, Keke Gai, Md Liakat Ali An Investigation on Cyber Security Threats and Security Models, 2015
[13] https://www.globalsign.com/en/blog/cybersecurity-trends-and-challenges-2018/
[14] Sophoslabs 2018 Malware Forecast https://www.sophos.com/en-us/en-us/medialibrary/PDFs/technical-papers/malware-forecast-2018.pdf?la=en
[15] A third of Americans live in a household with three or more smartphones, 2017.http://www.pewresearch.org/
[16] https://www.statista.com
[17] Mohammed J. Aljebreen Towards Intelligent Intrusion Detection Systems for Cloud Computing, 2018
[18] Zulaiha Ali Othman, Lew Mei Theng, Suhaila Zainudin, Hafiz Mohd Sarim Great Deluge Algorithm Feature Selection for Network Intrusion Detection. 2013
[19] Mafarja, M. and S. Abdullah, 2011. Modified great deluge for attribute reduction in rough set theory. Fuzzy Systems and Knowledge Discovery, pp: 1464-1469. 2011
[20] Chebrolu, S., A. Abraham, J.P. Thomas, 2005. Feature Deduction and Ensemble Design of Intrusion Detection Systems. Journal of Computers and Security, 24(4): 295-307.
[21] Zainal, A., M. Maarof, S. Shamsuddin, 2007. Feature selection using rough-DPSO in anomaly intrusion detection. Computational Science and Its Applications–ICCSA, pp: 512-524.
[22] Pratik N., Neelakantan, C. Nagesh M. Tech, 2011. "Role of Feature Selection in Intrusion Detection Systems for 802.00 Networks" International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) 1(1).
[23] Amr S Abed, T Charles Clancy, and David S Levy. Applying bag of system calls for anomalous behavior detection of applications in linux containers. In Globecom Workshops (GC Wkshps), 2015 IEEE, pages 1–5. IEEE, 2015.
[24] S Barlev, Z Basil, S Kohanim, R Peleg, S Regev, and Alexandra Shulman-Peleg. Secure yet usable: Protecting servers and linux containers. IBM Journal of Research and Development, 60(4):12–1, 2016.
[25] Qiang Wang Vasileios Megalooikonomou A Clustering Algorithm for Intrusion Detection, 2005.
[26] Martin Roesch Snort — Lightweight Intrusion Detection for Networks, Proceedings of LISA '99: 13th Systems Administration Conference Seattle, Washington, USA, November 7–12, 1999
[27] Nahla Ben Amor, Salem Benferhat, Zied Elouedi Naive Bayes vs Decision Trees in Intrusion Detection Systems, ACM Symposium on Applied Computing, 2004.
[28] Quinlan, J. R. C4.5, Programs for machine learning, Morgan K aufmann San Mateo Ca, 1993.
[29] http://www.netresec.com/?page=PcapFiles
[30] http://www.cybersecurity.unsw.adfa.edu.au/ADFA%20IDS%20Data sets/
[31] http://bit.ly/csic-2010-http-dataset_csv
[32] Haydar Teymourlouei, Lethia Jackson, 2017 How big data can improve cyber security, Proceedings of the 2017 International Conference on Advances in Big Data Analytics, pp: 9-13.
[33] Miltiadis Allamanis, Earl T. Barr, Premkumar Devanbu, Charles Sutton A survey of Machine Learning for Big Code and Naturalness, 2018.
[34] Robert Mitchell and Ing-Ray Chen A Survey of Intrusion Detection Techniques for Cyber-Physical Systems, 2014.
[35] Robert A. Bridges, Tarrah R. Glass-Vanderlan, Michael D. Jannacone and Maria S. Vincent A survey of Intrusion Detection Systems Leveraging Host Data, 2019
[36] Abiodun Ayodeji, Tong-Kuo Liu, Nan Chao, Li-qun Yang A new perspective towards the development of robust data-driven intrusion detection for industrial control sytems, 2020.
[37] Hadeel Alazzam, Ahmad Sharieh, Khair Eddin Sabri A Feature Selection Algorithm for Intrusion Detection System Based on Pigeon Inspired Optimizer, 2020.
[38] Chaouki Khammassi, Saoussen Krichen A NSGA2-LR wrapper approach for feature selection in network intrusion detection, 2020.
[39] Faezah Hamad Almasoudy, Wathiq Laftah Al-Yaseen, Ali Kadhum Idrees Differential Evolution Wrapper Feature Selection for Intrusion Detection System, 2019.
[40] Amol Borkar, Akshay Donode, Anjali Kumari A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and Protection System (IIDPS), 2017.
[41] Said Ouiazzane, Malika Addou, Fatimazahra A Multi-Agent Model for Network Intrusion Detection, 2019.
[42] Manoj s. koli, Manik K. Chavan An Advanced method for detection of botnet traffic using Internal Intrusion Detection, 2017.
[43] Azzedine Boukerche, Lining Zheng, Omar Alfandi Outlier Detection: Methods, Models and Classification, 2020.
[44] Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung Intrusion Detection Using Neural Networks and Support Vector Machines, 2002.
[45] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham Principle Components Analysis and Support Vector Machine-based Intrusion Detection System, 2010.
[46] Fidalcastro. A, Baburaj. E. Sequential Pattern Mining for Intrusion Detection System with Feature Selection on Big Data, 2017.
[47] Shi-Jie Song, Zunguo Huang, Hua-Ping Hu and Shi-Yao Jing. A Sequential Pattern Mining Algorithm for Misuse Intrusion Detection, 2004.
[48] Pakinam Elamein Abd Elaziz, Mohamed Sobh, Hoda K. Mohamed Database Intrusion Detection Using Sequential Data Mining Approaches, 2014.
[49] Wenke Lee and Salvatore J. Stolfo Data Mining Approaches for Intrusion Detection, 1998.
[50] Zhendong Wu, Jingjing Wang, Liqing Hu, Zhang Zhang, Han Wu A network intrusion detection method based on semantic Re-encoding and deep learning, 2020.
[51] Arwa Aldweesh, Abdelouahid Derhab, Ahmed Z. Emam Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A survey, Taxonomy and Open Issues, 2020.
[52] Omar Y. Al-Jarrah, Yousof Al-Hammdi, Paul D. Yoo, Sami Muhaidat, Mahmoud Al-Qutayri Semi-supervised multi-layered clustering model for intrusion detection, 2017.
[53] Naila Belhadj Aissa, Mohamed Guerroumi A Genetic Clustering Technique for Anomaly-Based Intrusion Detection Systems, 2015.
[54] Mohammad Khubeb Siddiqui and Shams Naahid Analysis of KDD CUP 99 Dataset using Clustering based Data Mining, 2013.
[55] Joshua Oldmeadow, Siddarth Ravinutaka and Christopher Lechie Adaptive Clustering for Network Intrusion Detection, 2004.
[56] S.Sathya Bama, M.S.Irfan Ahmed, A.Saravanan Network Intrusion Detection using Clustering: A Data Mining Approach, 2011.
[57] M. Mazhar Rathore, Anand Paul, Awais Ahmad, Seungmin Rho, Muhammad Imran, Mohsen Guizani Hadoop Based Real-Time Intrusion Detection for High-speed Networks, 2016.
[58] Sanraj Rajendra Bandre, Jyoti N. Nandimath Design Consideration of Network Intrusion Detection System using Hadoop and GPDPU, 2015.
[59] Sanjai Veeti and Qigang Gao Real-time Netwok Intrusion Detection Using Hadoop-Based Bayesian Classifier, 2014.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 01, January-2021

[60] Sandhya Peddabachigari, Ajith Abraham, Johnson Thomas Intrusion Detection Systems Using Decision Trees and Support Vector Machines, 2007.
[61] Shilpashree. S, S. C. Lingareddy, Nayana G Bhat, Sunil Kumar G Decision Tree: A machine Learning for Intrusion Detection, 2019.
[62] Manish Kumar, Dr. M. Hanumanthappa, Dr. T. V. Suresh Kumar Intrusion Detection System Using Decision Tree Algorithm, 2012.
[63] Christopher Kruegel and Thomas Toth Using Decision Trees to Improve Signature-Based Intrusion Detection, 2003.
[64] Kajal Rai, M. Syamala Devi, Ajay Guleria Decision Tree Based Algorithm for Intrusion Detection, 2015.
[65] Jianwu Zhang, Yu Ling, Xingbing Fu, Xiongkun Yang, Gang Xiong, Rui Zhang Model of Intrusion Detection System Based on the Integration of Spatial-Temporal Features, 2019.
[66] Iman Sharafaldin, Arash Habibi Laskkari and Ali A. Ghorbani Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 2018.