Classification of Malware Attacks Using Machine Learning In Decision Tree

Abel Yeboah-Ofori *u0118547@uel.ac.uk*
School of Architecture, Computing & Engineering *yeboah007@hotmail.com*
University of East London.
London, E16 2GA, UK

Abstract

Predicting cyberattacks using machine learning has become imperative since cyberattacks have increased exponentially due to the stealthy and sophisticated nature of adversaries. To have situational awareness and achieve defence in depth, using machine learning for threat prediction has become a prerequisite for cyber threat intelligence gathering. Some approaches to mitigating malware attacks include the use of spam filters, firewalls, and IDS/IPS configurations to detect attacks. However, threat actors are deploying adversarial machine learning techniques to exploit vulnerabilities. This paper explores the viability of using machine learning methods to predict malware attacks and build a classifier to automatically detect and label an event as "Has Detection or No Detection". The purpose is to predict the probability of malware penetration and the extent of manipulation on the network nodes for cyber threat intelligence. To demonstrate the applicability of our work, we use a decision tree (DT) algorithms to learn dataset for evaluation. The dataset was from Microsoft Malware threat prediction website Kaggle. We identify probably cyberattacks on smart grid, use attack scenarios to determine penetrations and manipulations. The results show that ML methods can be applied in smart grid cyber supply chain environment to detect cyberattacks and predict future trends.

Keywords: Cyberattack, Malware, Machine Learning, Smart Grid, Decision Tree.

1. INTRODUCTION

The unpredictable nature of cyberattacks and the cascading effects of cybercrimes on the business system have made it difficult for organizations to predict endpoint attacks. ML assist in recognizing attack patterns using datasets of previous attacks to predict future attacks trends and responses [1]. Endpoints are the third-party vendor systems, workstations, servers, handheld mobile devices and AMI devices. Malware attacks have intensified by the distributed nature of the smart grid in supply chain systems. Adversaries are using cyberattacks such as cross site scripting, cross site request forgeries, session hijacking and remote access trojan attacks to commit cybercrimes such as modification of software, manipulating of online services, manipulations electronic products, diverting e-products and other security misconfigurations. Ford and Siraj 2015, highlighted different issues in the applications of machine learning in cybersecurity by detecting phishing, network intrusion, testing security properties of protocols and smart energy consumptions profiling [2].

Machine learning techniques are applied in a cybersecurity environment to predict network intrusions detections, malicious codes detections, amount of suspicious transaction, electric power fraud anomaly detection, substation location frauds, and spam filtering for spear phishing attacks, as well as determine the probabilities of attacks. We could use ML to detect anomalies in HTTPs requests such as XXE, XSS, SSRF attacks in communication networks, authentication bypass in password setting and SQL Injection in a database system. Soska and Christin 2014, applied ML techniques to automatically detect vulnerable websites before the turn malicious [3]. Canali et al. 2014 applied ML techniques to detect the effectiveness of risk prediction based on browsing behaviours [4]. Hinks et al. 2015 use ML techniques on various classification algorithms to learn dataset to detect power system disturbance and cyberattack discrimination [1]. Mohasseb et al. 2019 applied ML techniques to analyze a dataset from various organizations to improve classification accuracies [5]. These works are important and contribute to detecting and predicting cyberattacks using machine learning in the cybersecurity domain. However, there is a limited focus on smart grid vulnerability from supply chain perspective, and specifically on threats relating to inbound and outbound chain contexts that need adequate detection to improve smart grid security control and decision makings.

In this paper, we use ML techniques to learn datasets and build a classifier to automatically detect and label an event as Has Detection or No Detection. The rationale for choosing the DT algorithm is that DT represents the major supervised schemes for ML in network security. We use a dataset from Microsoft malware prediction [6] for our work. To demonstrate the effectiveness of our approach, we adopt the decision tree algorithm to evaluate our data sets based on the attack classifications.

The main contribution of this paper is threefold. Firstly, we identify probably cyberattacks on the smart grid and the vulnerable sports that could be exploited through penetration and manipulations base on the telemetry dataset. Secondly, we use attack scenarios to determine the penetration and the manipulations for the threat predictions on the endpoint nodes. Finally, we use ML techniques to learn the dataset and use the DT algorithm to predict whether the endpoint nodes can classify if the nodes can detection cyberattack or not using Has Detection or No Detection. The results show that ML algorithms in Decision Trees methods could be applied in smart grid supply chain predictive analytics to detect cyberattacks and predict future trends.

The rest of the paper is structured as follows: Section 2 presents an overview of related works in the machine learning in smart grid supply security domain and the existing classification algorithms. Section 3 considers our approach to evaluating the ML techniques to learn dataset and the classification algorithms for smart grid supply chain, CPS smart grid infrastructure and the vulnerable spots and probable attacks scenarios. Further, it discusses the data representation, feature descriptions and extractions as well as the classification algorithm. Section 4 presents the implementation of the machine learning simulation process, performance evaluation on the classifier and determines the average accuracy of the model and predict the probability of penetrations on the endpoint nodes. Section 5 presents the results and analysis of the DT that predicts the cyberattack initiated and the cybercrimes committed or not. Further, we provide discussions of the several observations identified in the study. Finally, section 6 presents a conclusion of the study, comparisons of existing works, limitations and future works.

2. RELATED WORKS

This section reviews related works and the state of the art of cybersecurity in machine learning predictions, decision tree classifications and how they are related to malware attacks on CPS environment. That includes identification of previous classification approaches, leveraging the classifications of malware with a specific data set and prediction task used. Sharmar et al. 2012 proposed an ML technique for detecting worm variants of known worms in real-time systems [7]. Tsai et al. 2009, proposed a review of the intrusion detection system by using ML techniques and various classifiers on the intrusion detection domain [8]. Wang et al 2014. Performs an empirical study of adversarial attacks against ML models in the context of detecting malicious crowdsourcing systems [9]. Bilge et al. 2017, proposed a risk teller system that predicts cyber incidents by analyzing malicious files and infection records according to the endpoint protection software installed to determine machines that are at risk [10]. Canali et al. 2014 performed a correlation analysis on the effectiveness of risk prediction based on user browsing behaviour by leveraging ML techniques to provide a model that can be used to estimate the risk class of a given user [4]. Barros 2015 posits that decision threats and induction methods in general, arose in machine learning to avoid acquisition bottleneck for expert systems [11]. Villano, 2018, proposed a method of classification of internet logs using ML techniques by correlation and normalization process and evaluated the DT algorithm that could predict an attack or not [12]. Soska & Christin 2014, proposed a complementary approach to automatically detect vulnerable websites before they turn malicious by design, implement and evaluate a novel classification system which predicts whether a given website could be compromised in future [3]. Hinks et al. 2014, proposed an ML technique for power system disturbance and cyberattack discrimination by evaluating various ML methods for an optimal algorithm that is accurate in its classifiers to predict disturbance discriminators and implications [1]. Yavanoglu et al. 2017, proposed a review of cybersecurity datasets for ML algorithms by analyzing network traffic and detecting abnormalities used for experiments and evaluation methods considered as baseline classifiers for comparisons [13].

2.1 Decision Trees

Decision Tree is used as a method in ML for classification and regression in large and complex data in order to discover patterns. DT is built as a tree structure to classify instances of an attack by plotting each malware attack attributes from the top and down to its root. Each branch of the dataset is broken down into subsets to represent a choice of possible values for the attributes of output, and each leaf represents a decision. DT is used in supervised ML for classification and regression [11]. DT inference process starts at its root and proceeds to the leave. DT processes include splitting, pruning and tree selection [14]. Splitting includes partitioning the data into subsets, pruning includes the process of reducing the tree by turning some branch nodes into leaf nodes, and tree selection involves finding the smallest tree that fits the data. Each attribute is assigned a node, and in the leaf are the probable outcome or state. DT uses inductive inference as a method to arrive at a conclusion based on the independent input and using the dependent values as attributes. There are several approaches to DT algorithms such as J34, C4.5, C5.0 cart and others. We used C5 method to identify which attribute was the root of the tree [15]. Figure 1 shows an example of a DT.

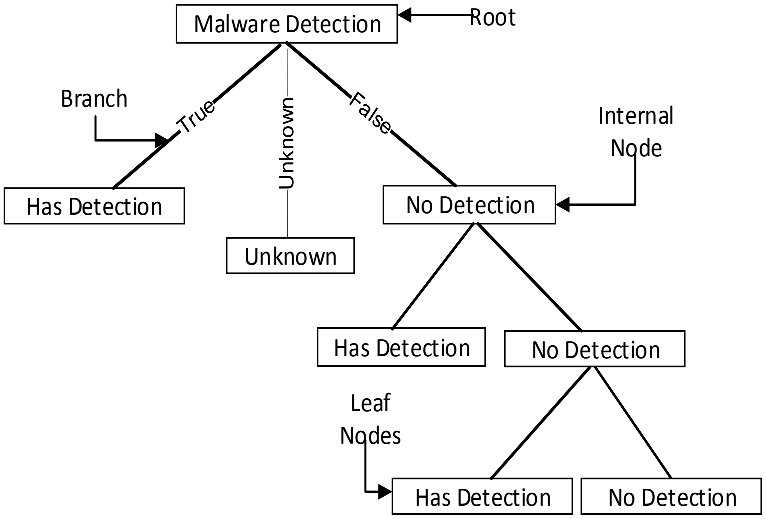

FIGURE 1: Decision Tree.

2.2 Decision Tree Selection Criteria

Decision Tree uses various algorithms for inferences to arrive at a conclusion, therefore, it is required to have a selection criterion with certain characteristics that we can use to determine the challenges. The characteristics include: Challenges originating from the classification of the data sets could be numerous. For instance, identifying attacks that are initiated through Intelligence Electronic Devices on smart grid systems or classifying staff salaries based on qualifications, skills and experience.

2.3 Rational for Chosen Machine Learning and Decision Tree

Several algorithms have been used in ML. such as Naive Bayes, SVM, Random Forest and Logistic Regression. However, our rationale for chosen DT algorithms in ML is that it provides discrete outputs as the factors are provided by attributed value pairs for strategic management decision makings. E.g. Results: Pass or Fail. Cyberattack: Internal or external. Temperature: hot or cold. Outcome: Positive or Negative.

- DT algorithm can identify attributes pairs that were not considered initially in the classification such as the source of attack but could work without those attributes to minimize inferred errors.
- DT algorithm can handle datasets that have errors in the attribute values and resolve classification errors in the training and test phase. Such as false positives (FP) output when network traffic is a normal or false negative (FN) when network traffic is under cyberattack. The discrete probability outputs provide results that predict a 'True or False', 'Yes or No' and 'A or B) outcomes.

3. APPROACH

This section considers our approach to evaluating the ML techniques to learn dataset and the classification algorithms for the smart grid supply chain. We discuss the smart grid infrastructures and the vulnerable spots, attack scenarios and the ML approach. The rationale for the ML approach using DT to predict an attack is to determine the causal relationships amongst the cyberattacks on a smart grid supply chain system and attempt to predict the malware using probability distribution methods. Then based on the classification analysis, we evaluate the predictive method with appropriate metrics to verify the organizational goal and security goal as we seek to determine whether a specific cyber threat phenomenon is likely to appear in a similar event. There are some algorithms for building decision trees such as ID3 and C4.5 formula and others [15]. We discuss the ML methods used, as well as the approaches used for the malware prediction.

3.1 CPS Smart Grid Infrastructure

The CPS smart grid infrastructure in figure 1, integrates application and network systems using Intelligence Electronic devices (IEDs). Refer IEC 61850 [16] The application system uses the IEDs, Sensors, Actuators and other communication devices for power generation, distribution, and transmission. The Supervisory Control and Data Acquisition (SCADA) and Programmable Logic Controls (PLC) establishes communication protocols with the Remote Telemetric Units (RTUs) for monitoring and gathering real-time data across various substation. The network system provides interconnectivity between substations, automation systems and field devices such as AMI and Home Energy Management Systems (HEMS) software [17] [19].

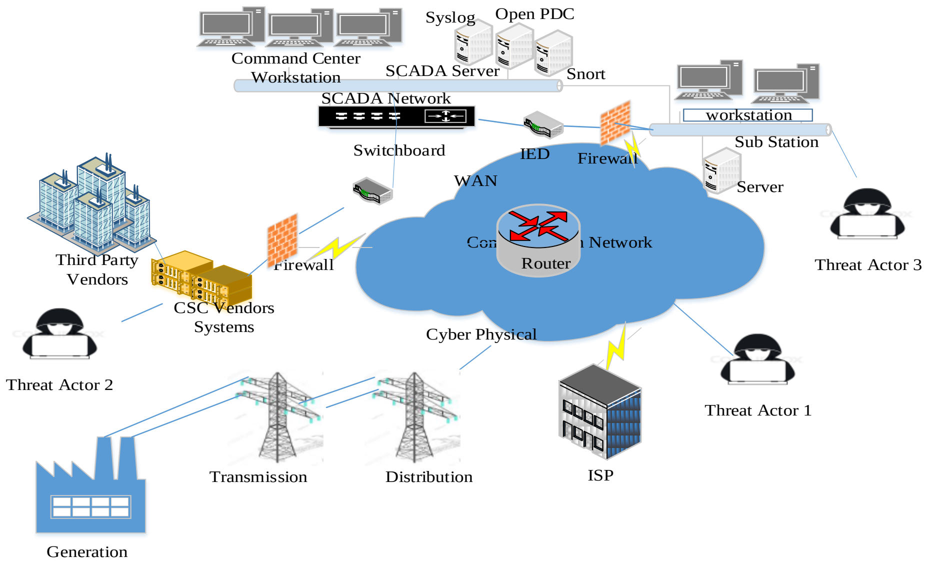

FIGURE 2: Smart Grid System Infrastructure and Vulnerable Spots [19].

The adversary could cause cyberattacks (penetration) and cybercrimes (manipulation) on the CPS. Cyberattacks such as remote access trojan, spear phishing, cross site scripting or session hijacking on the intelligent devices and communications networks to penetrate firewalls, IDS/IPS or the IEDs. After penetrating the system, the adversary could commit cybercrimes by manipulating the system to cause resonance attacks, DDoS attacks, IP theft, ID theft, intellectual property theft as well as take command and control to monitor and control the core business processes and operations. We include these attack scenarios in the analysis to determine the validity of the penetration and manipulations in real-time.

3.2 Attack Scenarios

We identify various attack scenarios for the study that will assist in the feature selection process as follows.

- Network Attack: An XSS or session hijacking attack on the CSC network may provide access to alter the smart metering system, change configurations using distance protection scheme to bypass controls in order to manipulate the software in the meter, prevent the system from recording accurate purchases or billings.
- Spyware Attack: The attacker could insert spyware or deploy a ransomware attack remotely to shut the systems down when the antivirus is outdated, and the software is unpatched, and consequently affect the prepaid card settings change the configurations using distance protection scheme so an attacker can manipulate and prevent accurate readings from valid purchases.
- Ransomware Attack: The attacker could use reconnaissance and social engineering tactics to gather intelligence and subsequently initiate a spear phishing attack on targeted users to shut the system down until a ransom is paid.
- Software Manipulation Attack: Most organizations fail to change the hard-coded password after buying software off the shelf. The attacker could deploy session hijacking techniques to exploit this vulnerability using advanced persistent threats and command & control techniques to manipulate the system and consequently cause cybercrimes such as intellectually property theft, ID theft and industrial espionage.
- DDoS or Data Injection: Attacker deploys DDoS attack that could consequently cause voltage surges by inserting a rootkit into the OS server to cause resonance attack on the smart grid components for the power system to oscillate.
- Island Hopping attack: On the CSC systems, vendors are more susceptible to cyberattacks, and the perpetrators are using RAT and Island-hopping attacks to gain access to the major organizations on the supply chain.
- Malware: The attacker could insert malware or spyware in the software that is bought off the shelf that gives the developers access to the system whenever users are prompted to update their software. That may cause software errors and subsequently lead to application system manipulations.

3.3 Threat Prediction Scenarios

The threat prediction attempts to investigate two kinds of scenarios that will determine the classification result. The scenarios use's ML techniques to determine the cyberattack initiated and the cybercrime committed based on the scenarios and the cyberattacks.

- Scenario 1: what is the probability of the penetration on the endpoint nodes?
- Scenario 2: What is the extent of manipulation on the various endpoint node?

3.4 Analytical Approach

To determine the viability of using ML techniques to learn dataset for penetrations and manipulations on the CPS, we used the DT algorithm and open source data from Microsoft Malware Predictions endpoint protection solutions website [3]. DT provides an efficient and nonparametric method that can be applied to a classification or regression task. Further, we used supervised learning to train and test the dataset as it provides an accurate prediction of system performance. Using DT hierarchical data structures for supervised learning provides input space that is split into local regions in order to predict the dependent variable for decision makings [11].

3.5 Data Representation

The data represented Microsoft Windows Machine's probability of getting infected by various families of malware, based on different properties of that machine. The telemetry data containing threat report were collected by Microsoft Windows Defender from various MS windows operating systems [6]. The properties and machine infections were generated by combining various user activities on different organization and vendors. The dataset we used contains 4000 entries. DT algorithm used determines attributes that return the highest information gain that satisfies the four uncertainty axions in a confusion matrix and provides the degree of disorganization in the dataset. Further, an Entropy formula was used to determine the information gained and the degree of uncertainty by separating the positive and negative rates as follows:

$$Entropy\ (E) = -\ a\ log_1\ a - b\ log_2\ b \qquad (1)$$

3.6 Feature Extraction

The feature extraction process involves removing irrelevant columns names or duplicates in the dataset to have unique values when training the data. Columns with a higher number of duplicates are removed to the correct data. The command is to count the 62 variables and remove the irrelevant variables. The output prints 62-8 = 54. (8 columns removed). However, the 4000 datasets were maintained. The classifier is set to model the features based on the importance as well as the F Score. The F-Score was used as the harmonic mean to determine the combinations of the precision and recall for plotting the model.

3.7 Classification Algorithm

The classification phase involves using the ML algorithm to test the dataset for prediction. In this phase, the DT model was used to split the data for prediction to determine if each endpoint node can detect infections or not. We considered C4.5 or C5 algorithms [15]. The training data is used to build the DT model, and the test data is used to determine the dimensionalities of the dataset. The rationale for choosing the DT algorithm is that DT represents the major supervised schemes for ML in network security. We train the ML algorithm using the training sets. Then compare the performance of the algorithm over the datasets.

4. IMPLEMENTATION

This section discusses the implementation of the machine learning simulation process. The purpose of the study is to use the DT algorithm to predict cyberattack and indicate it as Has Detection or No Detection. The dataset and the machine malware infections were gathered by Microsoft Defender endpoint protection [6]. The dataset corresponds to a machine identifier that provides results as to whether the Microsoft endpoints can predict if it can detect malware attacks on the nodes. As discussed in section 2, the DT algorithm learns from data sets to approximate an 'if then else' decision rules and generate branches for the tree nodes and decision nodes. We follow the process below to build the DT classifier for our prediction.

4.1 Description of Data

The dataset is about a malware attack on Microsoft Endpoint system and such systems can be a critical part of the smart grid CSC systems overall business continuity [6]. The dataset was designed to meet certain business constraints in relation to privacy and time periods in which machine was used. CSC integrates various organizational systems for the business process and information dissemination in the CPS environment. The data set containing these properties and the machine infections were generated by combining threat reports collected by Microsoft Endpoint Protection Solution, Windows Defender. Each row in the dataset corresponds to a machine unique identified by a Machine Identifier. Further, the dataset was created to meet certain business constraints, both regarding privacy and when the machine was running. Hence, the dataset is relevant for our work as it was gathered from global machines that used Microsoft

Windows Defender. The rationale for using the dataset for our work is that the dataset does not represent Microsoft customers machine only as it has been sampled to include a much larger proportion of malware machines. Thus, we used the dataset for our work to determine whether the has detection or no detection on various network nodes for threat predictions. Below are some of the features from the metadata that are relevant for our work [6].

- MachineIdentifier - Individual machine ID
- GeoNameIdentifier - ID for the geographic region a machine is located in
- DefaultBrowsersIdentifier - ID for the machine's default browser
- OrganizationIdentifier - ID for the organization the machine belongs in.
- is protected - This is a calculated field derived from the Spynet Report's AV Products field.
- Processor - This is the process architecture of the installed operating system
- HasTpm - True if the machine has tpm
- over - Version of the current operating system
- OsBuild - Build of the current operating system
- Census_DeviceFamily - AKA DeviceClass. Indicates the type of device that an edition of the OS is intended for desktop and mobile
- Firewall - This attribute is true (1) for Windows 8.1 and above if windows firewall is enabled, as reported by the service.

4.2 Data Preparation

The dataset represents Microsoft malware prediction events collected from various families of malware infections based on different properties of attacks. Windows Defender tool was used to generate the threat reports of the malware infections from various Microsoft endpoint protection solutions. [6]. The dataset derived 4000 entries with 64 columns, and each row represents different metadata entry. Each row in the dataset corresponds to a machine uniquely identified by a Machine Identifier. We used supervised learning to derive the dataset that represented the instance of each table and attribute. The rationale is to predict an outcome for future events. The variables in the datasets are for each instance to determine whether a malware attack is Has Detection or No Detection.

4.3 Feature Selection

The features for the dataset are split into the partition of the subsets of attacks as indicated in table 1. The attack features indicate the categories of attacks grouped. The splitting of the attack categories builds the classifications model for the three structure and breaks it down to represent the attack features. Further, we pruned the dataset to reduce the size of the tree by turning some branches nodes into leaf nodes. For instance, we categorized the attacks based on the threat descriptions in the table for us to fit the training data for the classifier and finds the tree that produces the lowest cross validation.

Attack Category	Attack Features	Threat Descriptions for Probable Cause of Attack
1	XSS/Session Hijacking	Default Browser vulnerabilities and injecting code in the URL or website
2-5	Spyware/Ransomware	Outdated Antivirus/Patches that are not updated regularly
6-7	Spear Phishing	Use Reconnaissance to identify vulnerable spots and attach email with a virus
8-9	Session Hijacking	Exploit Unchanged Hard-Coded password in software bought off the shelf
10-14	Rootkit/DDoS	Attack on BIOS or attach a virus to a USB key to cascade when booting.
15-20	RAT/Island Hopping	Attacks from Vendor systems to gain access to the organizational system
21-28	Ransomware/Malware	Exploiting outdated OS versions and encryptions especially TLS/SSL

29-35	Malware/Spyware	Packet injection and Resonance attacks
36-38	DDoS	Exploit IP Address Systems and Packet injections

TABLE 1: Attack Category and Feature Descriptions.

4.4 Performance Evaluation on the Classifier

The performance evaluation on the classifier determines the average accuracy of the models when we run the integer values in the cell. The performance of the model will be determined on the following values: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) rates. Further, the FPR and FN will be determined based on the elements.

$$F = 2\,\frac{Precision\ .Recall}{Precision+Recall} \tag{2}$$

5. RESULTS

In this section, we present the analysis of the investigation of threat prediction to the two scenarios for the classification results. We discuss adversarial ML briefly and how adversaries use ML techniques to exploit vulnerabilities. As discussed in section 3.3, the scenarios use the DT algorithm to predict the cyberattack initiated and the cybercrime committed.

5.1 Determine the Accuracy of the Threats

For us to predict the probability of an attack, we need to determine the known and the unknown attacks. As listed in Table 1, Known attacks include Malware, Spyware, Ransomware and, RAT, Cross-Site Scripting, Session Hijacking, Cross Site Request Forgery. These are the known attacks that could be identified. However, unknown attacks are cyber crimes committed after the attacks. Here, after gain access using the known attacks, the attacker, using APT and C&C to commit cybercrimes such as manipulation during development, manipulation during development, altering and changing delivery channels. The extent of these cybercrimes manipulations and the cascading impact are unknown and unquantifiable.

Scenario 1: Predict the probability of penetration on the endpoint nodes?

Determining the accuracy process involves evaluating the threats, and its impacts on the various network nodes for understanding and to provide cyber threat intelligence of the causes and effects cyberattacks on the organizational goal, the business process, financial impact. Table 2 presents the performances accuracies of the DT classifier of each cyberattack on various endpoints of the network. Using the confusion matrix, we determine the harmonic mean between the Precision (P), Recall (R) and F-Score (F). From the table, XXS/Section Hijacking, spear phishing, RAT/Island Hopping attacks predicted a higher probability of the penetration on the endpoint nodes with a percentage score of 82%, 75% and 75% respectively. However, the results revealed the XSS and Session Hijacking are the most like penetration method to deploy base on the predictions.

SCENARIO	DT			PREDICTIONS
ACCURACY	83%			100%
CYBERATTACKS	P	R	F	RESULTS
XSS/Session Jacking	0.89	0.41	0.75	82%
Spyware/Ransomware	0.89	0.58	0.85	87%
Spear Phishing	0.81	0.37	0.71	75%
Session Hijacking	0.71	0.39	0.64	65%
Rootkit/DDoS	0.66	0.37	0.68	55%
RAT/Island Hopping	0.67	0.30	0.74	68%
Ransomware/Malware	0.89	0.55	0.71	85%
Malware/Spyware	0.87	0.58	0.78	84%
DDoS	0.78	0.36	0.65	66%

TABLE 2: Predicting the Probability of Penetration.

Scenario 2: Predict the extent of manipulation on the various endpoint node?

Predicting the extent of manipulations on the various endpoint nodes after penetrations are very challenging due to the invincibility, uncertainty and fuzzy nature of cybercrimes. Further, determining the extent of cyberattack propagation and manipulations in an integrated network environment posse a major challenger in the cybersecurity discipline. From Table 2: the results indicate that:

- Ransomware, Malware and Spyware predicted a higher probability of manipulations on the endpoint nodes with a percentage score of 87%, 85% and 84% respectively after determining the Precision and F-Score with a low Recall rate.
- That indicates that the extent of manipulation in a given event could be high with an average accuracy of 85%.
- The manipulations could result in cyberattacks such as Industrial Espionage, Intellectual property theft, Advanced Persistent Threat and Command & Controls.

6. DISCUSSION

Predicting cyberattacks in real-time is challenging due to factors such as type of OS being used, system refresh rates, time zones, running updates and data in transition. Attacks such as Ransomware or malware may impact on the system based on the OS being used, the origin of the attack and due to the time zone. Threat actors could use adversarial machine learning techniques to exploit vulnerabilities in ML threat predictions

6.1 Adversarial Machine Learning

Adversarial machine learning is a technique used by the adversary to inject malicious input data in the dataset during the training and testing phase to manipulate the classification algorithms for the model. The technique can be used in supervised learning algorithms for cybersecurity datasets to exploit vulnerabilities and compromise performance results of malware detections, spam filters and IDS/IPS intrusions when predicting cyberattack trends and predicting the probability of fraudulent activities. The adversary could cause an increase in the false-positive rates by inserting malicious samples in the test phase to generate wrong classifications rates of the sample data. The adversarial machine learning technique could be used to manipulate training data to violate security policy, gain knowledge of threat intelligence, adversary capabilities and level of manipulations.

6.2 Determining Processor Count for Vulnerable Operating System

The classification of the malware attack is built based on the type of operating system that is being used by the organization. The OS determines the nature of antivirus that can be installed and could be exploited on each system and if it can detect malware attacks or not. An outdated antivirus within a third-party system could easily be a point of failure if a malware attack is initiated from there leading to power loss, power surge, system error or power fluctuations issues. Addressing downtime and uptime in the event of failure is critical for all the organizations that are integrated on the supply chain. For instance, a redundant array of independent disk (RAID) uses multiple hard drives in unique groupings and storage capacity mechanisms to produce a storage solution that provides improved throughput, resistance, and resilience. These drivers rely on antivirus updates and patches as trusted sources to prevent any compromises. Figure 3 explains the ML processor count of the systems and how it determines the speed at which the malware attack could occur as well as the extent of propagation in the event of an attack. The X-axis determines the process count for the vulnerable OS. The Y-axis determines the number of OS that are affected. From Figure 3, we realized that the processor count of 3000 was able to affect 4.0 systems. Indicating that the region identifier may have fewer systems with a higher probability of penetration and manipulation. Thus, exploiting outdated OS versions and encryptions especially TLS/SSL raises antivirus protection issues and application interoperability on the various network nodes. Users can be lured into a false sense of security by the threat actor across deferent platforms to update the antivirus that could lead to heuristic detections or false positives.

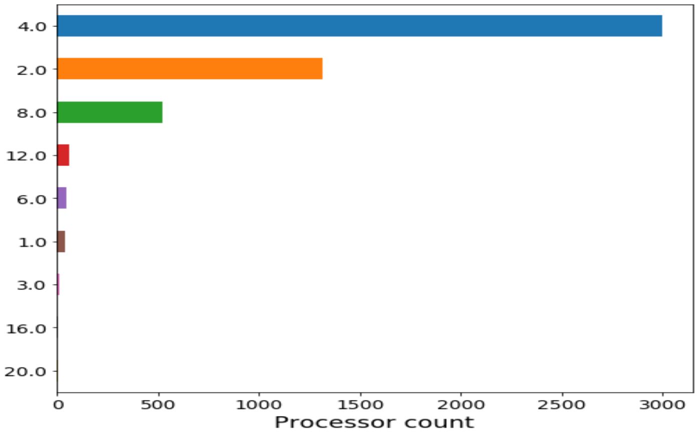

FIGURE 3: Processor Count for Vulnerable Operating System.

6.3 Resolution Rate of Probable Ransomware Attack

The figure displays the detection rate of the training dataset using the feature description of smart screen usage in the CSC environment. The implication of the of our results is that attackers have used malware attacks to penetrate the smart screen and inset spyware in the system that turns the camera on in the smart screen monitor. With that, the attacker can see everything the victim will do, take command and control, leading to cybercrimes attacks such as Intellectual property and industrial espionage. Refer to the FLocker mobile ransomware attack on smart screens (Duan. 2016). Further, the attacker could use social engineering tactics such as spear phishing to cause ransomware attacks to deface the monitors and cascade to other smart screen systems on the CSC.

Ransomware attacks could affect CSC system platforms that use multiple smart screens monitors by infecting a single screen and may propagate to others on the monitors with the same network nodes and lock the screens during run time. Section 4.1 describes the dataset and how each row in the dataset corresponds to a machine unique identified by a Machine Identifier gathered from global machines that use Microsoft Windows Defender. The FLocker ransomware infects smart screens and avoids detections as the code is always being rewriting to improve its routine variants and meet changing trends. When launched, the malware identifies the country ID, the machine ID and activates depending on the motives and intents of the adversary. Figure 4 identifies the vertical resolution rates of the various systems and how the infections propagate through the systems during run time. The Y-axis indicates the extent of vertical infections and the X-axis indicates the resolution rates of the infected systems. Malware or ransomware that is embedded directly into the requested web page in the attack could propagate to other systems.

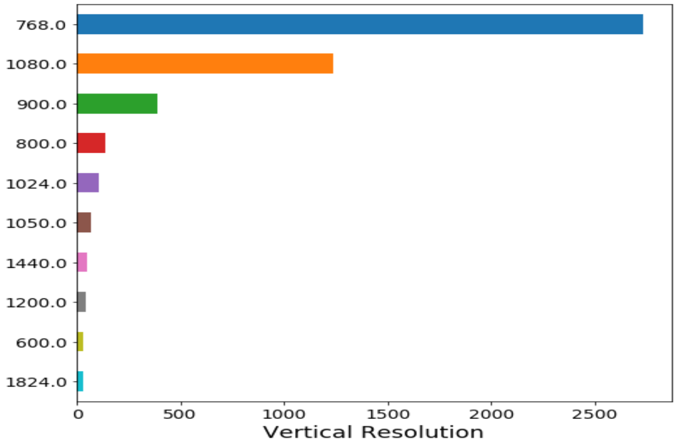

FIGURE 4: Resolution Rate of Probable Ransomware Attack.

6.4 Decision Tree Predictions

The DT in Figure 5 depicts the results of the classifier that predict a Windows machine probability of getting infected by attacks listed in table1, based on various properties of that machine. The properties used to generate the DT are, SmartScreen, CountryIdentifier, AVProducts, OSInstallTypeNname, TotalPhysivalRam, OsBuildLab, OSWUAutoOptionsName. The Smart screen represents workstations. The Country identifier represents the country the Windows Operating System is located. According to a report by Controller and Audit General on the investigations of 'WannaCry' Ransomware Attack in 2017, the attack initially infected the NHS system the UK and then propagated to other countries across the world and infected various system [18]. The report indicated that the OS antivirus product was outdated hence the attack. The objective of the paper is to use ML techniques on a dataset to predict whether the system can detect an attack and label that as Has Detection or No Detection. From, the sample dataset of 4000, we predict the probability of ransomware attack infection based on the type of OS Installed that could lead to the vulnerability of the ransomware infecting the smart screen as well as the country the OS is installed and the version.

6.5 Gini Index Based Decision Tree

The Gini index based decision tree was the calculation for Smart Screen Malware (M) Infection Trend.

- If the dataset (D) contains examples from n class, then the Gini index, gini(D) is defined as:

$$Gini(D) = 1 - \sum_{i=1}^{n} (P_i)^2 \qquad (3)$$

Where p_i is the probability of an object being classified to a particular class that infected.

- If a dataset (D) is split on the root (R) into two sets subsets D_1 and D_2 the gini index (D) is expressed as:

$$\mathrm{Gini(D)} = \frac{D_1}{D} gini(D_1) = \frac{D_2}{D} gini(D_2) \qquad (4)$$

The Reduction in Impurity for the split in the dataset was calculated as:

$$\mathrm{Gini(R)} = \mathrm{gini(D)} - \mathrm{gini}_R\,(\mathrm{D}) \qquad (5)$$

From the DT algorithm, we calculate the information gained after the malware (M) infection trend test is applied on the smart screen for the classification. A weighted sum of Gini Indices was calculated using the DT and generated the Has Detection and No Detection tree.

Figure 6 depicts the DT indicating the results of the gini index used to measure the probability of infections of a ransomware attack that may be wrongly classified. The DT root indicates a smart screen rate of <= 6.5 with a split Gini of 0.5 indicating an equal distribution of the dataset. The root of the three has an initial dataset of 4000 as the sample size. The DT algorithm split the value into two sets: [1973, 2027]. From the analysis, 1973 were identified as has detection, hence are not vulnerable to the attacks. However, 2027 were found to have no detection hence vulnerable to malware or ransomware attacks. The branch with has detection is indicated as (True) and the other with no detection is indicated as (False). identified as has identified as a country identifier with the class Has detection, identified the values of 2531 and 1648 from a sample size of 3531. A sample size of 2955 has antivirus product installed. However, the total physical RAM has no detection rate of 1458. The DT split the sample size further till the values were at the threshold. Figure 5 depicts the gini index calculated and information gained after the DT test is applied.

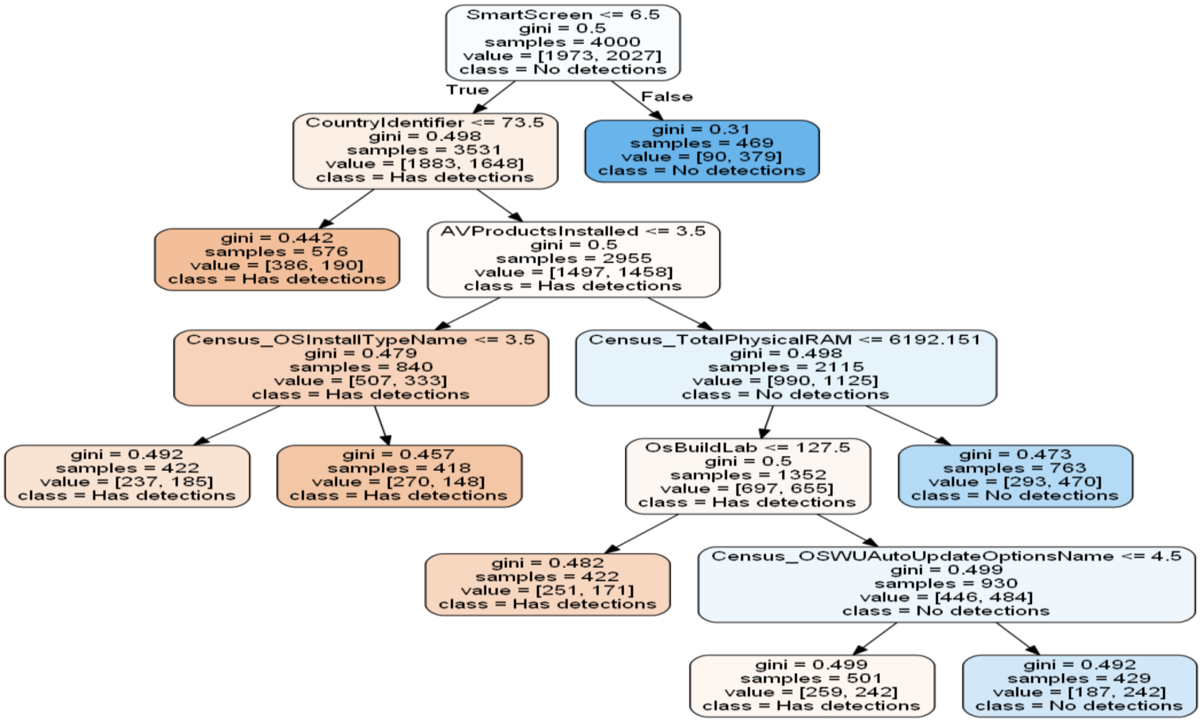

FIGURE 5: Decision Tree Predictions.

The results from Scenarios 1 and 2 provides cyber threat intelligence as to what could happen in the event of a cyberattack without the classifications of the detections rates in Figure 5.

Scenario 1 predicted a higher probability of the penetration's attacks on the endpoint nodes after determining the harmonic mean between the Precision, Recall and F-Score with a percentage score of:

- 82% for XXS/Section Hijacking
- 75% for spear phishing
- 75% for RAT/Island Hopping attacks

Scenario 2, determining the extent of cyberattack propagation and manipulations in an integrated smart grid network environment. The results show that cyberattacks such as Ransomware, Malware and Spyware predicted a higher probability of attack propagation and manipulations to other systems.

- The results indicate the extent of manipulation to other integrated network systems could be high with an average accuracy of 85% in a given event.
- The extend of manipulations indicates the relevance of the classification of the cyberattack. The threat intelligence indicates that it could result in cyberattacks such as Industrial Espionage, Intellectual property theft, Advanced Persistent Threat and Command & Controls.

6.6 Comparing Our Results with Existing Work

A significant amount of literature exists in machine learning techniques and classification algorithms to learn dataset for performance accuracies in the cybersecurity domain that have considered threat predictions. Comparing our results to existing works, we considered works that used ML methods from cyberattack penetration and cybercrimes manipulations perspective to detect attacks. Hinks et al. 2014, considered an ML technique for power system disturbance and cyberattack discrimination by evaluating various ML methods for accurate classification to predict disturbance discriminators and implications [1]. Sharmar et al. 2012 use ML to detect worm variants of known worms in real-time [7]. Bilge et al. 2017 applied ML techniques to predict a risk teller system for cyber incidents by analyzing malicious files [10]. Canali et al. 2014 used the ML method to perform a correlation analysis of the effectiveness of risk prediction based on user browsing behaviour [4]. Villano, 2018, applied ML method for correlation and normalization process and evaluated the DT algorithm that could predict an attack or not [12]. Soska & Christin 2014 used the ML approach to automatically detect vulnerable websites [10]. Mohasseb et al. 2019 applied ML approach for predictive analytics using SVM and Naïve Bayes algorithms for evaluation accuracies [5].

Further, various DT algorithms, models and techniques have been implemented using a various dataset for building intrusion detections, anomaly detection and threat predictions. Pournouri et al (2017) proposed a cyber attack analysis using decision tree techniques to learn an open source intelligence dataset for prediction and for improving cyber situational awareness [21]. Patel and Prajapati (2018) proposed a study and analysis of decision tree based classification using ID3, C4.5 and CART algorithms to learn a dataset to determine the best performance accuracy [22]. Moon et al (2017) proposed an intrusion detection system based on a decision tree using analysis of attack behaviour information to detect the possibility of intrusion for preventing APT attacks [23]. Sarker et al (2020), presented a machine learning intrusion detection system based security model called "IntruDTree" that evaluated various algorithms on a dataset by ranking the security features according to their importance then build a generalized tree for detecting intrusions [24]. Das & Morris (2018) presented a survey of machine learning and data mining methods for cybersecurity applications and analytics for intrusion detection and traffic classifications in emails by evaluating the various classifications algorithms on a dataset for performance accuracies [25]. Balogun & Jimoh (2015) proposed a hybrid of DT and KNN algorithms to detect anomaly intrusions [26]. Malik et al (2018) used a hybrid of DT pruning and BPSO algorithms for network intrusion detection [27]. Rai et al (2016) proposed C4.5 DT algorithm to construct a model for intrusion detection [28]. Yeboah-Ofori & Boachie (2019) present a malware attack predictive analytics using various ML Classification algorithms in a majority voting for performance accuracies [29]. Ingre et al (2017) proposed a DT algorithm that classifies an IDS dataset as normal or attack after the learning and testing the dataset [30]. Relan and Patil (2015) used a variant of C4.5 DT algorithm to implement an IDS by considering discrete values for classifications [31].

However, none of the works explored the viability of using machine learning methods to predict malware attacks and build a classifier to automatically detect and label an event as Has Detection or No Detection on smart grid supply chain domain to predict the probability of penetration and the extent of manipulation on the network system nodes for cyber threat intelligence and situational awareness.

7. CONCLUSION

Our work focused on using ML to learn dataset and used the DT algorithm to determine whether the classifier can predict an attack and label the attack as Has Detection or No Detection. In this paper, we have used a malware prediction dataset from a well-known source learn the dataset. We have used the DT algorithm to model the infections. Although, other algorithms can perform the same task that the DT could handle datasets that have errors in the attribute values and resolve classification errors in the training and test phase. Based on our result, the precision was 83% accurate and concluded that supervised learning model performed better in our predictions. Description of objects may include attributes based on measurement or subjective judgement, both of which might give rise to errors in the values of the attributes. Some of the objects in the training set may even have been misclassified. Take, for instance, a malware attack classification rule from a collection of cyberattacks events. An attribute might test for the presence of propagation of attack that might give a positive or negative reading at some point. However, questions remaining to be addressed as to what performance evaluation methods could provide the best performance indicators for threat predictions and cyber threat intelligence gatherings that could provide security control mechanisms. There are limitations in our work, such as comparing other classification algorithms for predictive analytics due to the invincibility nature of cyberattacks and the cascading impacts on other system nodes.

Future Works

Future research will focus on using ML techniques on various classification algorithms to learn the dataset for anomaly detection and to predict cyberattacks trends. The approach will assist to determine the best performance metrics, for cyber threat intelligence and predict future trends.

8. REFERENCES

[1] C. R. B. Hink, J. M. Beaver, M. A.. Bukner, T. Morris, U. Adhikari S. Pan. "Machine Learning for Power System Disturbance and Cyber-attack Discrimination" 7th International Symposium on Resilient Control Systems. IEEE Xplore. 10.1109/ISRCS.2014.6900095. (2014).

[2] V. Ford. A. Siraj. "Application of Machine Learning in Cyber Security". Conference Paper. Computer Science Department. Tennessee Tech University. (2014).

[3] K. Soska, N. Christin. "Automatically Detecting Vulnerable Websites Before They Turn Malicious. In Proceeding of the 23rd UNENIX Security Symposium. Carnegie Mellon University. ISBN 978-1-931971-15-7 (2014).

[4] D. Canali, L. Bilge, D. Balzarotti. "On the Effectiveness of Risk Prediction Based on User Browsing Behaviour". ACM 978-1-4503-2800-5/14/06. http://dx.doi.org/10.1145/2590296.2590347. (2014). [Accessed 20/04/2020].

[5] A. Mohasseb, B. Aziz, J. Jung, and J. Lee, "Predicting Cyber Security Incidents Using Machine Learning Algorithms: A case study of Korean SMEs". University of Portsmouth Research Portal. (2019).

[6] Microsoft Malware Prediction. Research Prediction. (2019). (https://www.kaggle.com/c/microsoft-malware-prediction/data). [Accessed 26/01/2020].

[7] O. Sharma, M. Girolami J. Sventek, "Detecting Worm Variants using Machine Learning". DOI: 10.1145/1364654.1364657 (2007).

[8] C. Tsai, Y. Hsu, C. Lin, W. Lin. "Intrusion detection by machine learning: A review Expert Systems with Applications". 36.10, pp. 11994-12000, (2009).

[9] G. Wang. T. Wang. H. Zheng, B. Y. Zhao. "Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers". In *Proceedings of the 23rd USENIX Security Symposium* San Diego, CA, pp. 239–254, (2014).

[10] L. Bilge, Y. Han, M. D. Amoco, Risk Teller: Predicting the Risk of Cyber Incidents. ACM ISBN 978-1-4503-4946-8/17/10. https://doi.org/10.1145/3133956.3134022 CCS (2017). [Accessed 14/12/2019].

[11] R. C. Barros, A. c. P. L. F. De Carvalho. A. A. Freitas, "Automatic Design of Decision-Tree Induction Algorithms", Springer. Briefs in Computer Science, DOI 10.1007/978-3-319-14231-9_2. (2015).

[12] E. G. V. Villano. "Classification of Logs Using Machine Learning". Norwegian University of Science and Technology. (2018).

[13] O. Yavanoglu. M. Aydos. "A Review of Cyber Security Dataset for Machine Learning Algorithms". International Conference on Big Data, IEEE Xplore. DOI: 10.1109//BigData.2007.8258167. (2018).

[14] A. Boschetti. L. Massaron. “Python Data Science Essentials”. 2nd Edition. UK. ISBN 978-1-78646-213-8. (2016).

[15] J. R. Quinlan. “C4.5: Programs for Machine Learning”. 16, 2333-240 Department of Computer, John Hopkins University, Baltimore. MD21218. (1994).

[16] W. Wang, Z. Lu, “Cyber Security in Smart Grid: Survey and Challenges”. Elsevier. (2013).

[17] A. Yeboah-Ofori, S. Islam. “Cyber Security Threat Modeling for Supply Chain Organizational Environments”. Future Internet, 11, 63, doi: 10.3390/611030063, (2019).

[18] Controller and Audit General: Investigation. “Wannacry Cyber-attack and The NHS”. Department of Health. National Audit Office. UK (2017).

[19] A. Yeboah-Ofori. Islam, S. Brimicombe A: Detecting Cyber Supply Chain Attacks on Cyber Physical Systems Using Bayesian Belief Network. International Conference on Cyber Security and Internet of Things. (2019). DOI 10.1109/ICSIoT47925.2019.00014.

[20] Duan, E. (2016). FLocker Mobile Ransomware Crosses to Smart TV. Trend Micro. Security Intelligence Blog. https://blog.trendmicro.com/trendlabs-security-intelligence/flocker-ransomware-crosses-smart-tv/ [Accessed 10/03/2020].

[21] S. Pournouri, B. Akhgar, P. S. Bayerl. “Cyber Attacks Analysis Using Decision Tree Techniques for Improving Cyber Situational Awareness” International Conference on Global Security, Safety and Sustainability. Springer. Vol.360. 2017. DOI: 10.1007/978-3-319-51064-4_14.

[22] H. Patel, P. Prajapati. “Study and Analysis of Decision Tree Based Classification Algorithms” International Journal of Computer Science and Engineering. 2018. DOI: 10.26438/ijcse/v6i10.7478.

[23] D. Moon, H. Im, I. Kim, J. H. Park. “DTB-IDS: An Intrusion Detection System Based on Decision Tree Using Behavior Analysis for Preventing APT Attacks” Springer, The Journal of Supercomputing 73 2881-2895. 2017. DOI: https://doi.org/10.1007/s11227-015-1604-8.

[24] I. H. Sarker, Y. B. Abushark, F. Alsolami, A. I. Khan. “IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Systems” MDPI. Symmetry 12, 754, doi:10.3390/sym12050754.

[25] R. Das, T. Morris. “Machine Learning in Cyber Security”. IEEE Xplore. International Conference on Computer, Electronic and Communication Engineering. 2018. DOI: 10.1109/ICCECE.2017.8526232.

[26] A. O. Balogun, R. G. Jimoh. “Anomaly Intrusion Detection Using in Hybrid of Decision Tree And K-Nearest Neighbor”. Journal of Advances in Scientific Research & Application. 2015.

[27] A.J. Malik, F. A. Khan. “A Hybrid Technique Using Binary Particle Swarm Optimization and Decision Tree Pruning for Network Intrusion Detection”. Cluster Computing. 21, 667–680. 2018. doi.org/10.1007/s10586-017-0971-8.

[28] K. Rai. M. S. Devi, A. Guleria. “Decision Tree Based Algorithm for Intrusion Detection”. International Journal Advanced Networked Applications. Vol 7, Issue 04. Pages: 2828. 2016.

[29] A. Yeboah-Ofori, C. Boachie. “Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning” IEEE Explore. CSIoT pp. 66-77 2019, doi: 10.1109/ICSIoT47925.2019.00019.

[30] B. Ingre, A. Yadav, A. K. Soni “Decision Tree Based Intrusion Detection System for NSL-KDD Dataset”. International Conference on Information and Communication Technology for Intelligent Systems. 25–26, pp. 207–218. 2017.

[31] N. G. Relan. D. R. Patil. “Implementation of Network Intrusion Detection System Using Variant of Decision Tree Algorithm”. IEEE Xplore. International Conference on Nascent Technologies in the Engineering Field. pp. 1–5. 2015. DOI: 10.1109/ICNTE.2015.7029925.