Malware attack predictive analytics in a cyber supply chain context using machine learning

Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning

Abel Yeboah-Ofori
School of Architecture, Computing & Engineering
University of East London. UK
u0118547@uel.ac.uk

Charles Boachie
Center for Biostatistics
University of Aberdeen. UK
charlieboa@hotmail.com

Abstract: **Due to the invincibility nature of cyber attacks on the cyber supply chain (CSC), and the cascading effects of malware infections, we use machine learning to predict attacks. As organizations have become more reliant on CSC systems for business continuity, so are the increase in vulnerabilities and the threat landscapes. Some traditional approach to detecting and defending malware attack has largely been antimalware or antivirus software such as spam filters, firewall, and IDS/IPS. These tools largely succeed, however, as threat actors get more intelligent, they are able to circumvent and affect nodes on systems which then propagates. In our previous work, we characterized threat actor activities, including presumed intent and historically observed behaviour, for the purpose of ascertaining the current threats that could be exploited. In this paper, we use ML techniques to learn dataset and predict which CSC nodes have detection or no detection. The purpose is to predict which modes are venerable to cyberattacks and for predicting future trends. To demonstrate the applicability of our approach, we used a dataset from Microsoft Malware Prediction website. Further, an ensemble is used to link Logistic Regression, and Decision Tree and SVM algorithms in Majority Voting and run on the training data and then use 10-fold cross-validation to test the parameter estimation, accurate results and predictions. The results show that ML algorithms in Decision Trees methods can be used in cyber supply chain predict analytics to detect and predict future cyber attack trends.**

Keywords: Machine Learning, Cyber Supply Chain, Predictive Analytics. Cyber Security. Cyberattack

I. INTRODUCTION

The cyber supply chain system is very effective and efficient from a business process point of view as it ensures information flows, reduced inventory and service time to deliver [1]. However, from a cybersecurity standpoint, the CSC system is critically vulnerable due to the various connected network hosts and nodes. The CSC is a highly integrated network as it provides access to (1) various organizational services such as electronic commerce, online banking, distribution, and delivery (2) sensitive data such as Intellectual property, customer data, and financial data. Ensuring confidentiality, integrity, and availability of the supply chain system has been a major challenge facing organizations in maintaining security on online services and information flows. The CSC systems are more vulnerable due to its integrated and distributed nature. It uses public facing IPs for the various organizations and third party vendors on the supply chain system and that makes it accessible by attackers. Recent studies have found that over 90% of Web hosts have serious vulnerabilities [2, 3] and 60 of the 100 most popular websites hosted or were involved in the malicious activity on supply chain systems**.** The malware attacks made possible on the supply chain are numerous and varied, but generally involve attackers that are either injecting a virus, a worm, a bug or a Trojan into software or in an HTTP request that could execute on the network and the on the webserver when processing a request. Attackers could execute arbitrary commands on the supply chain systems remotely and that could cascade to other systems onto other nodes on the network. The attacker can then manipulate the vulnerable spots and maintain Advance Persistent Threats (APT) and Command & Control attacks. For the dataset, we use a publicly available data source from a Microsoft Malware Prediction website [4].

The aim of this paper is to use ML techniques to learn dataset and predict which CSC nodes have detection or no detection. The purpose is to predict which modes are venerable to cyberattacks and for predicting future trends. To demonstrate the applicability of our work, we used Logistic Regression (LR), Decision Tree (DT) and SVM algorithms to test the data classifications.

The novelty contributions are: (1) to used ML to analysis and understanding cyber supplier chain attacks by using cross-validation test on LG, SVM, and DT algorithms to generate true values to determine the algorithm that produces the better predictive performance. (2) Combined the three algorithms within Majority Voting (MV) to determine which of them produced the highest accuracy. The results generate a DT for our ML predictive analytics that predicts a has detection of no detection. The results show that ML algorithms methods can be used in CSC security to predictive analytics to detect and predict future cyber attack trends.

II. RELATED WORKS

In this section, we review related works and the state of the art in cybersecurity, machine learning and decision tree and how they are related to malware attacks on CSC systems. This includes identification of previous classification approaches, leveraging the classifications of malware with a specific data set and attack prediction tasks used. The following are the related works:

A. Machine Learning

Machine Learning (ML) technics are used to learn dataset to classify algorithms and for accurate predictions [5]. The purpose of using ML is to use previous cyberattacks to predict future attacks and make informed decisions [6]. Machine learning could be applied in smart grid cyber supply chain security environment to predict electric power fraud anomaly detection, the amount of suspicious transaction, substation location frauds,

978-1-7281-7417-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSIoT47925.2019.00019

network intrusion, and spam filtering for spear phishing attacks, as well as determine the probabilities of attacks. CSC systems include linking together of different organizational websites that align their business processes, goals, objectives, and some components of their systems to third party organizations, suppliers, consumers, and partners. We could use machine learning cybersecurity to detect anomalies in HTTPs requests such as XXE, XSS, SSRF attacks in communication networks, authentication bypass in password setting an SQL Injection in the database. Gallagher et al. 2009, develop a TTP attack classifier based on the vector space model used commonly for information retrieval. The authors used a machine learning approach to build a classifier to automatically label as request Valid or Attack. The authors approach for dealing with HTTP-based attacks is to identify malicious code in incoming HTTP requests and eliminate bad requests before they are processed. They further demonstrated their approach through experiments on the ECML/PKDD 2007 Discovery Challenge data set [7]. Hinks et al. 2014, explored the suitability of ML methods as a means of discriminating power systems disturbances. The authors evaluated various ML methods including OneR, NNge, Random Forest, Naïve Bayes, SVM, JRipper, Adaboost, in disturbance discrimination and the practical implication. They theorized that ML algorithms will leverage the non-linear complex relationship between power systems measurements and as to sufficiency to discriminate between non-malicious and natural disturbance [8]. Buczak et al 2016, carried out a survey that describes a focused literature survey for machine learning and data mining methods for cyber analytics in support of intrusion detection in cybersecurity applications. The authors discussed cybersecurity data sets and provided comparison criteria for machine learning and data mining methods for recognition of types of the attack (misuse) and for detection of an attack (intrusion) [9]. However, the techniques and methods used are not ML and DT. Sharma et al. 2007, reported on the feasibility of using ML techniques to detect variants of known worms in real-time. The authors applied the SVM algorithm in standard pattern recognition in the work to worm detection problems. They investigated the optimal configurations of SVM and associate kernels functions to classify various types of synthetically generated worms. The used linear kernel to demonstrate the results using unnormalized bi-gram frequency counts as input [10]. Yavanoglu et al. 2017, proposed a dataset used in artificial intelligence and machine learning techniques which are the primary tools for analyzing network traffic and detecting abnormalities. The authors compared the machine learning techniques used for experiments, evaluation methods considered and baseline classifiers for comparison [11]. Villano, 2018, investigated the decision tree algorithm and its implementation in WEKA software. The author researched the processes of correlation and normalization of logs. The author further evaluated the algorithm that could predict an attack or not after a training phase using internet logs and proposed a framework designed for the normalization and correlation processes [6]. Bhamare et al. 2016, analyzed the performance of major supervised learning algorithms with different datasets namely UNSW and ISOT obtained in a simulated cloud environment for cloud security. The authors compared Logistic Regression, DT, Naïve Bayes, and SVM classification algorithms and their techniques [12].

B. Supervised and Unsupervised Machine Learning

There are two types of ML: Supervised and Unsupervised: Supervised ML consist of working with a dataset that includes both input and outputs parameters. Supervised learning is able to provide an accurate prediction of system performance using the dataset for training and dataset for testing. ML uses two types of algorithms: classification and regression: The classification algorithm is used to identify the major features or class level of each object, depending on the class to which it corresponds to when it was defined in the beginning. The classification could be separated into binary or multiclass. We use binary variables to detect positive or negative values. Algorithms used include Logistic Regression, Support Vector Machine (SVM)., Decision Tree.

Regression algorithms are used to predict continues response values by utilizing knowledge of existing data to have an idea of the new data. It can be used to predict the cost of impact, asset value and cost of alternative and probability of fraudulent actions.

- Logistic regression is a classification algorithm in ML that is used to predict the binomial probability of a categorical binary variable [5] and estimates the relationship between one dependent variable and one independent variable.
- SVM algorithm learns from the data itself what distribution of the features should be and therefore it is applicable in a large variety of situation when you want to catch all the outliers but also the unusual data examples [5]. SVM methods support outliers detection and can be specific for decision functions. The input vectors are non-linearly mapped to a high-dimension feature space [13].
- Majority Voting (MV) algorithm used to verify that a prediction satisfies a majority in a given list of outcomes [5] determine the highest number or percentage representation in a list of algorithms.

Unsupervised ML is more of a data driving approach with the intention to find anomalies in data. They are used when there are no labelled data and model should somehow mark it by itself based on the properties. However, it works less precisely than that of supervised approach as the system is not provided with data sets and therefore used for predicting unknown outputs [6].

C. Decision Trees

Decision trees are an efficient nonparametric method that can be applied to classification or regression tasks. They are hierarchical data structures for supervised learning whereby the input space is split into local regions in order to predict the dependent variable [15]. Barros 2015, posits that Decision trees and induction methods in general, arose in machine learning to avoid acquisition bottleneck for expert systems. [16]. DT 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-like structure that classifies instances of an event by plotting each malware attack attributes from the top, down to its root. The figure below shows an example of a DT.

There are several approaches to ML classification and algorithms used such as LG, SVM, DT, Random Forest, Naïve Bayes, XBGoost, LightGBM, and CatBoost [5] that has been demonstrated as successful in the cybersecurity domain. We used LG, SVM, DT method to identify which attributes best fit the tree, determine the best classification algorithm and provide accurate cyberattack predictive analytics. 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 those attacks that are initiated through classifying staff salaries based on qualifications, skills, and experience.

III. APPROACH

This section considers the different ML approaches that can be used to solve cybersecurity tasks and how they are related to CSC system threats. The algorithms used include DT, LG and SVM in majority voting. The aim of this paper is to use ML techniques in a decision tree to predict which CSC system nodes have detection or no detection. The process includes data description, features Selection, choosing a classifier, performance evaluating and prediction as depicted in figure 1.

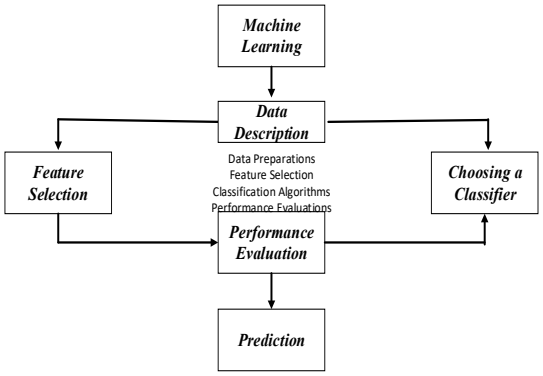

Fig 1. Machine Learning Process

A. Dataset Description

The dataset is about malware attack in Microsoft endpoint system and such a system can be a critical part of the CSC systems overall business continuity [4]. The dataset was designed to meet certain business constraints in regard to privacy and time period in which machine was used. CSC integrates various organizational systems for the business process and information dissemination in CPS environment. Hence, the dataset is relevant for our work as it was gathered from global machines that used Microsoft Windows Defender. The data set containing these properties and the machine infections were generated by combining threat report 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 in regard to the privacy and when the machine was running. 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. For the dataset, we use publicly available data from Microsoft Malware Prediction website [4].

B. Feature Extraction

Feature extraction assists in analysing the various classifications algorithms and ensure an accurate representation of the dataset. The extraction process involves using different techniques to select the available feature in the data for the application of the ML algorithms. The following are some features from the telemetry data relevant for our work.

- 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.

C. Choosing a Classifier

The classifications are implemented using ML algorithms such as LR, DT and SVM in MV. For our study, we use the binary classification as it supports AUC-ROC in distinguishing between the probabilities of the given classes. Further, its precisions can predict correct instances, provides a harmonic mean for precision, recall and F-score. The optimization algorithm is used to identify the major features or class level of each object. We used ensemble to combine the algorithms and test the dataset on each to determine the accurate prediction and best results. Further, a K-Fold classifier is used to run each algorithm 10 times for best results.

D. Performance Evaluation

To evaluate the performance of the ML algorithms, confusion metrics are used in ML classifications to determine actual or predictive values precision, recall and F-Score. These values include the True Positive (TP), True Negative (FP), False Positive (FP) and False

Negative (FN). The accuracy of the confusion metrics is the proportion of the total number of predictions that are considered as accurate. We use the following equation [6] below to determine the accuracy (AC) of the matrix.

Accuracy

$$Accuracy\ AC = \frac{\#\ of\ correct\ prediction\ (TP+TN)}{\#\ of\ predictions\ (TP+FP+TN+FN)} \quad (1)$$

True Positive Rate (TPR)

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

Precision (P) determined as correct the proportion.

$$P = \frac{TP}{TP+FP} \quad (3)$$

F-Score determines the harmonic mean of P and R.

$$F = 2\ \frac{Precision\ x\ Recall}{Precision+Recall} \quad (4)$$

The instances of the accuracies are based on the following classifications for the model prediction.

- TP = An instance that is classified as positive and predicted correctly as positive?
- TN = An instance that is classified as negative and predicted correctly as negative?
- FP = An instance that is classified as negative but predicted wrongly as positive?
- FN = An instance that classified as positive but predicted wrongly as negative?

IV. IMPLEMENTATION

This section discusses the implementation of the machine learning simulation process. The purpose of the study is to predict the probability of malware infections on Microsoft Windows. The dataset and the machine malware infections were gathered by Microsoft Defender endpoint protection [4]. The dataset corresponds to a machine identifier that provides results as to whether the Microsoft endpoints are able to predict if it can detect malware attacks on the nodes. Detecting malware on supply chain networks could be challenging as the various systems could respond differently due to its current status and time stamps. Some systems may be going through software updates, patches, upgrades and various configurations at various times. Hence, the implementation process used to generate the dataset may come with constraints. However, due to these challenges, we may experience different predictive results from our test data and in the cross-validation. Therefore, we used ensemble and combine three classification methods: Logistic Regression, DT and SVM for our machine leaving to determine the best results and the highest accuracy.

A. Data Preparation

The data was downloaded from the Microsoft Malware prediction (Kaggle) website [4]. The number of entries in the dataset we used was 20000. There were 62 attributes in the data sets and they have all been correlated. The data was loaded from a pre-prepared dataset by calling the categories of the machine learning identifier. Then using the print command that loads the training set from the folder we have created. The output generated 20000 training datasets with 62 variables. The data was prepared by converting the average of the columns of the dataset and set the command to return the columns, the number of floats and the mean. The command removes all the duplicates in the training set.

The output prints 62-8 = 54. (8 columns removed). However, the 20000 datasets were maintained. The output was:

B. Preparing Data for the Ensemble Test

The Ensemble was used as a method of bringing different algorithms together to test the dataset, *choose performance metrics* and determine their accurate prediction [5]. The tools and the algorithm used are to test the data classifications, prevent overfitting and provide better analysis and understanding of the algorithms. We use the ensemble to test the dataset using LR, SVM and DT algorithms in MV. The pipeline was used to link and run the data frames together. A 10-fold cross-validation test was performed on the algorithms by training the data. validate the parameter 10 times as the values could change when we run it only ones. GridsearchCV was used to provide an exhaustive search over specified parameter values for the estimator.

C. Choosing an Optimization Algorithm: Combining LG, DT, SVM in MV

Majority voting (MV) algorithm provides us with the ability to combine all the three algorithms in the classifiers to determine the mean score of the total results. The print should generate the ROC AUC for the percentage scores for the mean, standard deviation, and the label. The output is as below: 10-fold cross-validation:

- ROC AUC: 0.66 (+/- 0.02) [Logistic Regression]
- ROC AUC: 0.58 (+/- 0.02) [Decision Tree]
- ROC AUC: 0.66 (+/- 0.02) [SVM]
- ROC AUC: 0.66 (+/- 0.02) [Majority Voting]

D. Evaluating the performance of the Model: Plot Accuracy of Algorithms in ROC-AUC

To determine the performance of the model, we plot the run the algorithms in ROC. AUC_ROC (Area Under Curve – Receiver Operating Characteristics) uses a model selection metric for bi-multiclass classification problem to distinguish between the probabilities of the given classes. It determines the True Positives Rates and False Negatives Rates [5]. We label it to determine the x-axis as True Positive Rate and y-axis as False Positive rate. The output indicates that the DT has AUC of 0.59 which is not a good prediction the TPR and FPR as it las less detection rate comparatively. We plot the graph below:

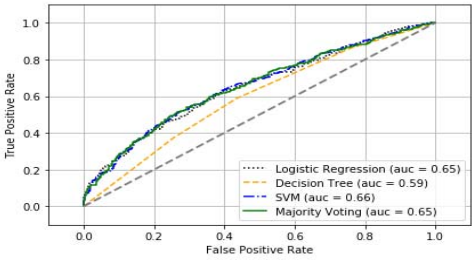

Fig 2. Plot the accuracy of all the algorithms in ROC

E. Tuning the Algorithm

We use K-Fold parameter tuning also to fine-tune and determine the best results comparatively. For each value, we take 4/5 of the training set as train and 1/5 as K-Fold test. We train and check the ROC results and take the min_sample_leave value that gives provides the best results. The min_sampe_leaf was set to [200, 400, 500, 600, 700, 800, 1000]. To increase the dataset to effect changes in the minimum sample graph in figure 3.

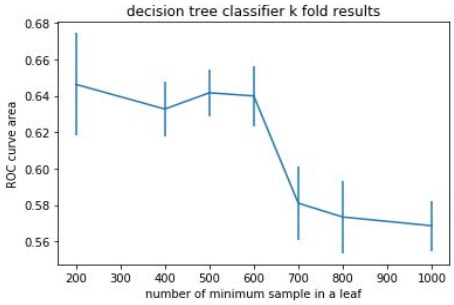

Fig 3. K-Fold Parameter Tuning

F. Display Size Correlation

G.

This section displays the correlation of the size of the screen to determine whether the machines have detections. It counts the number of antiviruses installed on the machines and determines the size of the display in inches against the detection rate. Our output below shows that the ML algorithm counted that 3526 machines have 5.0 antivirus installed on it. 1341 machine has 4.0 antivirus installed on it. 112 machine has 3.0 antivirus installed on it. 16 machine has 6.0 antivirus installed on it. 4 machine has 2.0 antivirus installed on it and 1 machine has 1.0 antivirus installed on it.

H. Display Detection Rate of the Training Datasets

We plot the grid histogram of the detection rate of the training dataset to determine with the size of the screen has a correlation with the display size. The graph displays the Count, Display Size in Inches and the Detection Rate. The count indicates the processing counts of machines from 0-70. The display's size in inches indicates the size of the processes from 0-80 and the Detection Rate are from 0.0-0.8. The graph shows that from 5-10 the display size in inches increased from the count of the machines 4 to 35 indicating a detection rate of about 0.45. However, from 11-12 inches the count of machines went up to about 55 with a detection rate of 0.65. Further the from 13-25 inches, the machines counted were up to 50 with a detection rate of between 0.6. Similarly, from 26-35 inches the machine counts were about 48 but increased from 40-50 inches with a detection rate of 0.6. Similarly, from 36-45 inches the machine counts increased to about 48 but decreased to 10 inches with a detection rate of 0.25. However, the inches increased from 46 to 65 with a count 65 and a detection rate of 0.58. The screen size is a feature that can be used to predict the probability of an attack. We assume the display increases or detection size. The screen size is relevant in our machine learning as the screen size has a correlation with the detection rate.

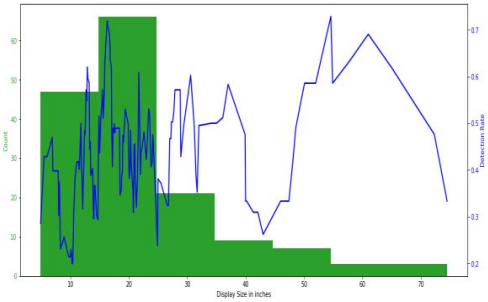

Fig 4. Display Detection Rate of the Training Datasets

I. Impact of Disk Capacity of on Detection Rates

The graph displays the total disk capacity from 0 to 4,000,000. As the disk capacity increases, the rate of detection could go up or down. Considering the nature of machines in use, the machine counts may increase from 0-98 with the total disk capacity from 0-500000 and an increase in the detection rate of 0.78. Conversely, the counts dropped to 22 on the total disk capacity from 500,000-550,000. However, from count 22 the total disk capacity increased from 1000,000-2000,000 with a detection rate of 0.5. Further, the count increased again to 60 on the total disk capacity of 2000,000 with a detection rate of 0.55. Similarly, the count dropped to 60 diagonally to 4000,000 on the total disk capacity with a detection rate of 0.55. These systems could be CPS such as industrial control system, smart grid, industrial plants, transport and communication systems with high integration and providing real-time services.

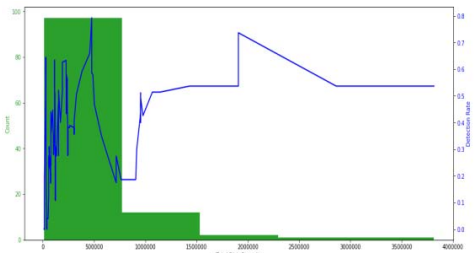

Fig 5. Impact of Disk Capacity of on Detection Rates

J. Determining the Detection

In this section, we compare the resolution ratio of each graph below to the detection rates to determine

whether the system Has Detection or No Detection. The Blue colour (0) has No Detection and the Orange colour (1) indicates Has Detection. The resolution rate affects the detection rates. For instance, the resolution ratio of 0.562 indicates that out of a total of 2500 counts, has No Detection is 1200 and Has Detection is 1300. Similarly, the second graph indicates that the resolution ratio of 0.5625 has no detection of 780 counts whilst has detection has 1000 making a total count of 1780. Further, the third graph has a resolution ratio of 0.625 with a No Detection of 220 and Has Detection of 125 indicating a total count of 345. Conversely, the fourth graph has a resolution ratio of 0.75 with a No Detection rate of 100 and Has Detection rate of 95 indicating a total count of 195. Furthermore, the fifth graph has a resolution ratio of 0.7998 with a No Detection of 50 and Has Detection of 50 indicating a total count of 100. Similarly, the sixth graph has a resolution ratio of 0.666 with a No Detection of 33 and Has Detection of 34 indicating a total count 64. Comparatively, the fifth graph has a low detection rate to the 6th graph and could be vulnerable to malware attacks on the nodes.

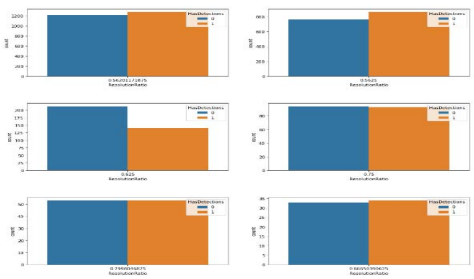

Fig 6. Resolution Ratio and Detection Rates

K. *Decision Tree Prediction*

The prediction and analysis of the ML and the results are projected the DT simulation. The variables for training and testing the algorithms are extracted from the dataset. The root of the tree is the SmartScreen feature set to <= 10.5. The three depicts a nominal variable with two arrows indicating True for 'Has Detection' False for 'No Detection' from the root. We set the Gini to 0.5 so that if the engine is true the variable goes left and if the Gini is false, it goes left on the plot as the information gained. The dataset is set to 5000 with a *minimum_sample_leaf* set to 200. The minimum_sample_leaf determines the depth of the tree. The value = [2464, 2536] is the summation of the dataset. The value 2464 predict No Detection on the machine nodes whereas the value 2536 Has Detection.

The AVProductsInstalled feature depicts a figure of <= 4.5, with a gain of 0.498. the sample size is 4414 and was split to determine the value which is [2361, 2063] with prediction class of Has Detection. Further, the AVProductsInstalled feature was split further as it predicted has detection. However, AppVersion_3 predicted no detection, therefore, the tree was not split further down.

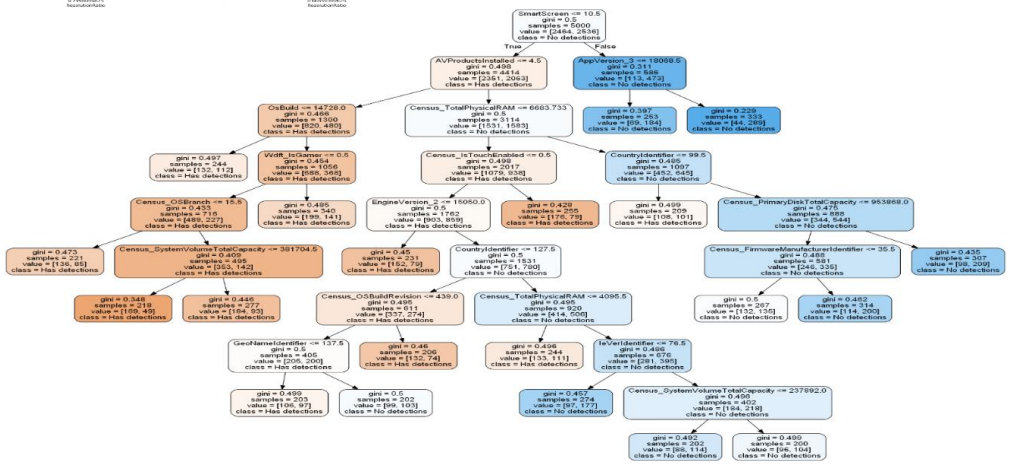

Fig 7. Decision Tree

L. *Malware Attack Prediction*

In previous work, [1], we characterized threat actor activities, including presumed intent and historically observed behaviour, for the purpose of ascertaining the current threats that could be exploited. Further, we identified eight vulnerable spots and their probability that the cyber attacker could exploit those spots namely the: Firewall, IDS/IPS, Vendors CSC system, Network, IP Addresses, Database, Software, and Websites. In this work, we use ML techniques to techniques and develop intelligent cyber threat techniques that can predict which nodes on our system are venerable to malware attacks to be able to predict future attacks.

To demonstrate the effectiveness of our approach, we used the results from Microsoft malware attack prediction to determine the probabilities of the malware attacks on the CSC nodes. The goal is to predict windows machines probability of getting infected by various families of malware attacks based

on the properties of the machine. We use the eight attacks in the previous work [REF] above and relate it to the predictions in the section for our results.

T. CSC Power System Framework Configurations

The figure below depicts the power system framework configurations used in the smart grid environment [1]. Tier 1 covers the transmission and distributions domains, using high-bandwidth communication media such as WiMAX and Fiber on a wireless area network (WAN). The IED monitors and control the electric power transmission to the distribution system using the phasor monitoring unit (PMU) for measuring instantaneous bus voltage, line current, and frequency. The command centre integrates with the SCADA system servers and uses a switchboard to establish communication with the IED units. Tier 2 provides a gateway for the Wireless Area Network (WAN) technologies and communication utilities to have access to the customers' premises for the CSC nodes. and demand response applications. It uses a collection of Intelligence Electronic Devices (IED) units to collect the various Phasor Measuring Units (PMUs). A malware attack could be initiated on the IDE and the network and could lead to attacks on the CSC, substation and the tier 1 infrastructures. Tier 3 integrates the local area network with the customer management systems (CMS) and uses the IED to communicate with the smart meter, which aggregates sensor information from various home appliance devices. We present the attack features in figure 8 and explain the predictions in Table 1 below.

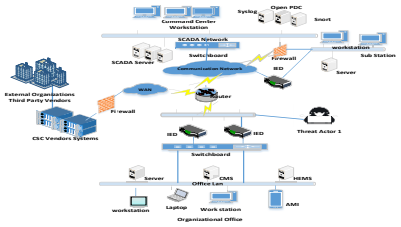

Fig 8. CSC Smart Grid

U. Type of Malware Predictions

The following are some of the attacks that could be initiated on the smart grid.

- Network Interruptions: remote attacks on the supply chain network nodes to cause interruption, interceptions, manipulation, and fabrications.
- Voltage Surges: a rootkit attack to cause resonance or DDoS attack on the components for the electric power to oscillate
- Software Errors: insert malware to infect software in order to cause application system manipulations and cybercrimes.
- Prepaid Card Setting Change: configurations are changed with a distance protection scheme to manipulate the card and prevent the card from accurate readings.
- Smart Metering Tampering: configurations are changed with a distance protection scheme so an attacker can manipulate the software in the meter so that the meter does not read for a valid purchase or valid outstanding.

Table 1. CSC Attack Features and Predictions

CSC Attack Features	Predictions
XSS/Session Hijacking	Default Browser vulnerabilities and injecting code in the URL or website
Spyware/Ransomware	Outdated Antivirus/Patches that are not updated regularly
Spear Phishing	Use Reconnaissance to identify vulnerable spots and attach email with a virus
Session Hijacking	Exploit Unchanged Hard-Coded password in software bought off the shelf
Rootkit/DDoS	Attack on BIOS or attach a virus to a USB key to cascade when booting.
RAT/Island Hopping	Attacks from Vendor systems to gain access to the organizational system
Ransomware/Malware	Exploiting outdated OS versions and encryptions especially TLS/SSL
Malware/Spyware	Packet injection and Resonance attacks
DDoS	Exploit IP Address Systems and Packet injections

V. RESULTS

The results of our implementation and our evaluation of the classifications of the various algorithms are as follows. The purpose of this paper is to use the ML techniques in a decision tree to predict which CSC system nodes have detection or no detection of malware. For us to have better predictive performances, we used several approaches for the ML classifications and algorithms such as LG, SVM, DT and MV methods to determine which algorithm provide the best predictive analysis attribute of the tree. The regression algorithms can predict continues response values by utilizing knowledge of existing data to have an idea of the new data. The classification algorithm was separated binomially. The features for each object was identify depending on the class it corresponds to when it was defined for the classifications. For us to estimate the best optimization over the specified parameter values for our estimator, we used GridsearchCV method to cross-validation the grid search in the parameter tunings. K-Fold cross-validation and a pipeline were used to link and run the various algorithms together for training the data to generate true values.

Further, we use AUC_ROC (Area Under Curve – Receiver Operating Characteristics) curve to predict the dimensions of the graph. The TPR is determined on the Y-axis and the FPR is determined on the X-axis. Furthermore, to determine the mean score of the total results, we used the MV algorithm to combine the LG, SVM and the DT in the classifiers and set the classification method to a random state. The output indicates that after testing the dataset five times on each algorithm, the ROC AUC results for LG was able to predict that LG has 0.66 probability of being attacked with a T or F value of 0.02. Similarly, the ROC AUC for DT was able to predict that DT has 0.58 probability of being attacked with a T or F value of 0.02. However, the ROC AUC for SVM was able to predict that DT has 0.66 probability of being attacked with a T or F value of 0.02. Therefore, our analysis reveals that DT was predicted to have a 58% higher probability of being attacked. Results also show the maximum depth of I

and C of 0.001 as the algorithm produced an accuracy of 66%. Finally, the DT depicts a tree that is able to provide ML with predictive analysis of machines nodes that Has Detections or No Detection.

A. Comparing Results Existing Works

There are several existing works on ML techniques for smart grid power supply environment for detecting malware attack prediction. For instance, Morris et al. 2014, used classification algorithms such as OneR, NNge, JRipper for rule inductions, Naïve Bayes for probability classification, SVM for the DT learning, Adaboost for boost a meta-algorithm for learning for their ML techniques. Further, Sharma et al. 2007, SVM algorithm in standard pattern recognition in the work for worm detection problems. Furthermore, Bhamare et al. 2016, compared Logistic Regression, DT, Naïve Bayes, and SVM classification algorithms and their techniques using WEKA. However, none of the authors, used LG, DT, SVM, RF in Majority Voting to compare their results. We used an ensemble to test the dataset. Further, we used Logistic Regression, Decision Tree and SVM algorithms to test the data classifications, prevent overfitting and provide better analysis and understanding. We used the pipeline to link the algorithms and run the training data and then use five-fold cross-validation to test the parameter estimation. We used GridsearchCV to estimate the best optimization to cross-validated the grid search in the parameter tunings. Further, we use AUC_ROC curve to predict the dimensions of the graph for the True Positives Rates and False Negatives Rates. The results show that ML algorithms and techniques can be used in cyber supply chain predict analytics to detect and predict future cyber supply chain attack trends.

VI. CONCLUSION

Machine learning has been used in the cybersecurity environment and has become the major tool to predict attacks dues to the invincibility, uncertainty and fuzziness in cyberattacks and the complicated and integrated nature of CSC systems. In this paper, we have used the ML techniques and develop intelligent cyber threat techniques that can predict which nodes on our system are venerable to attacks to be able to predict future attacks. To demonstrate the effectiveness of our approach, we used an ensemble to test the dataset. Further, we used Logistic Regression, Decision Tree, SVM and Random Forest algorithms to test the data classifications, to prevent overfitting and provide better analysis and understanding. We used the pipeline to link the algorithms and run the training data and then use five-fold cross-validation to test the parameter estimation. We used GridsearchCV to estimate the best optimization to cross-validated the GridsearchCV in the parameter tunings. Further, we use AUC_ROC curve to predict the dimensions of the graph for the True Positives Rates and False Negatives Rates. The tree predicts has detection or no detection. The results show that ML algorithms in Decision Trees methods can be used in cyber supply chain predictive analytics to detect and predict future cyber attack trends.

Future works will look at other ML approaches such as XGBoost, LightGBM, CatBoost algorithms that are relevant for cyber threat predictions in CSC environment. Further, deep learning-based approaches will be studied in cyberattacks prediction and cyber threat intelligence gatherings.

REFERENCES

[1] A. Yeboah-Ofori, and S. Islam. "Cyber Security Threat Modeling for Supply Chain Organizational Environments". Future Internet, 2019. 11, 63, doi: 10.3390/611030063.

[2] US-Cert. Building Security in Software & Supply Chain Assurance. Available online: https://www.us-cert.gov/bsi/articles/knowledge/attack-patterns (accessed on 20 September 2018).

[3] J. Freidman, and M Bouchard. "Cyber Threat Intelligence Guide: Using Knowledge About Adversary to Win The War Against Targeted Attacks." iSightPartners. 2018.

[4] Microsoft Malware Prediction. Research Prediction. 2019. (https://www.kaggle.com/c/microsoft-malware-prediction/data).

[5] A. Boschetti and L. Massaron. "Python Data Science Essentials". 2016. 2nd Edition. UK. ISBN 978-1-78646-213-8.

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

[7] B. Gallagher and T. Eliassi-Rad. "Classification of HTTP attacks: a Study on the ECML/PKDD 2007 Discovery Challenge". Lawrence Liverpool National Laboratory (LLNL) Livermore, CA. 2009

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

[9] A. L. Buczak, and E. Guven. "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection". IEEE Communications Surveys & Tutorials. Vol. 18. NO. 2, 2nd Quarter. 2016.

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

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

[12] D. Bhamare, T. Salman, M. Samaka, A. Erba and R. Jain. "Feasibility of Supervised Machine Learning for Cloud Security "3rd International Conference on Information Science and Security. Thailand. 2016.

[13] C. Cortes and V. Vapnik. "Support Vector Networks" Machine Learning. Vol, 20. No. 3. Pp. 273-297. 1995.

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

[15] E. Alpaydin, "Introduction to Machine Learning" MIT Press, Cambridge. 2010. ISBN:026201243X, 9780262012430.

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

[17] C. Shannon. "15-358 Probability and Computing. Elements of Information Theory. Lecture 25. 2009. http://www.cs.cmu.edu/~venkatg/teaching/ITCS-spr2013/notes/15359-2009-lecture25.pdf

[18] L. Rokach and O Maimon. "Top-Down Induction of Decision Trees Classifiers – A Survey" IEEE Transactions on Systems. MAN, and Cybernetics. Part C: Application & Reviews. Nov. 2005. Vol. 35. No. 4. 10.1109/TSMCC.2005.859799.