Machine Learning and Cyber Security

Rishabh Das
Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Huntsville, USA
rd0029@uah.edu

Thomas H. Morris, Ph.D.
Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Huntsville, USA
tommy.morris@uah.edu

ABSTRACT

The application of machine learning (ML) technique in cyber-security is increasing than ever before. Starting from IP traffic classification, filtering malicious traffic for intrusion detection, ML is the one of the promising answers that can be effective against zero day threats. New research is being done by use of statistical traffic characteristics and ML techniques. This paper is a focused literature survey of machine learning and its application to cyber analytics for intrusion detection, traffic classification and applications such as email filtering. Based on the relevance and the number of citation each methods were identified and summarized. Because datasets are an important part of the ML approaches some well know datasets are also mentioned. Some recommendations are also provided on when to use a given algorithm. An evaluation of four ML algorithms has been performed on MODBUS data collected from a gas pipeline. Various attacks have been classified using the ML algorithms and finally the performance of each algorithm have been assessed.

Keywords

Machine learning; Data mining; cyber security;

1. INTRODUCTION

This papers is a focused literature survey of machine learning and data mining methods for cyber security applications. Few ML methods are described along with their application in the field of cyber security. A set of comparison criteria for ML method is provided in the paper and a set of recommendations on the best method to use was made depending on the properties of the cyber security problems. Secondly, a MODBUS data set [1] has been used to compare the effectiveness of five different algorithms when applied to ICS networks. Receiver operating characteristic (ROC) is often used to choose optimal models and to discard sub-optimal one independently from the cost content or the class distribution. Hence, a ROC curve has been plotted to assess the performance of the binary classifier used with the data set under study.

This paper is intended for researchers willing to start their work in the field of ML and cyber security. Along with the description of the machine learning some references to prominent works have been cited and some valuable examples are put forth how cyber problems are often tackled by ML. From early 2000 several prominent surveys on the ML research has already been described. Nguyen et. al. [2] puts forth a comprehensive study of IP traffic classification technique that does not rely on well-known port numbers or known packet payloads. Techniques involving ML along with statistical traffic characteristics used in IP classification is reviewed in this paper. Nguyen et. al. reviewed 18 paper in this domain and is one of the most valued possession of any researcher starting their research in cyber security and ML related domains. Amomani et.al [4] puts forth an extensive survey about all the major e-mail filtering and ML techniques that can be used to classify and recognize phishing emails from normal ones. The state of the art research on such attacks have been enumerated and a comparative study of the all those techniques have been performed. Tedero et. al. [5] presents a statistical, machine learning and knowledge based approaches for network intrusion detection. It focusses mainly in the domain of anomaly detection and not on signature based detection. Filtering or classifying traffic on the fly is an important aspect that has been researched by a lot of cyber security personnel. Speroto et. al. [3] used NetFlow (Network Flow) data and proved that the packet processing may not be possible at streaming speed if the amount of network traffic is beyond certain threshold limit. These are significant works that outlines the current works related to ML and its application to the domain of cyber security and will be helpful to all researchers new to field of ML.

The contribution of the paper is to identify cyber security datasets that can be used by researchers and to point out the algorithms that can be applied to cyber specific problems. A set of machine learning algorithm have been evaluated in the later part of the paper on collected ICS dataset to identify various attacks while analyzing remote terminal unit (RTU) in a gas pipeline. The data set used has 35 different types of simulated attacks against ICS. The accuracy of each ML algorithm in the segregation of malicious traffic have been analyzed.

2. IMPORTANT CYBER SECURITY DATASET FOR MACHINE LEARNING

Data is of utmost importance in ML approaches. A researcher of machine learning has to understand the data set thoroughly before doing any kind of analysis. Secondly, raw data like packet capture (pcap), NetFlow and other network data is not directly usable in the ML analysis. The data has to be pre-processed to make it usable in popular ML tools like WEKA [6], R [7] and RapidMiner [8]. Hence researchers using ML analysis on custom system has to understand the data collection methodology and the methods that are used in preprocessing the data. This section will enlist few low level details on the data sets, and some popular tools used in capturing the data from the network.

2.1 Network Packet Data

There are a lot (144 as per Internet Engineering Task Force) of internet protocols that are used by programs running on the user levels. These protocols uses data packets as the main mode of communication. The network traffic in the form of packet received and transmitted at the interfaces (physical and wireless) can be captured and stored in the form of packet capture (pcap). Libpcap and Wincap for UNIX and Windows respectively are very popular network tools. Some tools like wireshark, tcpdump can also be used as protocol analyzer, packet sniffer and network monitor.

The dataset of machine learning has distinct features and attributes. These features defines the prime characteristics of each set of data in the dataset. Hence when a bulk of raw data is captured as pcap the researcher has to write some kind script to segregate the attributes needed from the pcap into ML tool usable format. Fowler et.al. [9] studied the attribute relation file format (arff) of Weka and developed a tool that can be used to convert any Packet Details Markup Language (pdml) format to weka minable arff format. To convert a pcap file to pdml tools like tshark can be used.

```
Tshark –T pdml –r <input file> < output file>
```

Where the input file is the .pcap file and the output file is the name of the pdml file. Secondly the Fowler's tool "pdml2arff.py" (available in GitHub) can be used to do perform the final conversion.

```
pdml2arff.py <Input file>
```

Where the input file is the name of the pdml file. This will generate an arff file called <Input file>.arff

Fowler's results shows that for normal tcp traffic the tool performs well and successfully converts the raw data into weka usable format. For the final analysis of the paper this tool was used with MODBUS protocol to convert the captured pcap files from the gas pipeline, it converted all the attributes into string nominal attributes which was readable by WEKA but was unusable for any kind of analysis. No numeric data was found in the arff file, hence Fowler's tool is not suitable for the MODBUS protocol.

A comprehensive list of packet headers of cyber security dataset of few important protocol have been enumerated by Buczak et. al [10].

2.2 Data from NetFlow

Cisco has its own feature called NetFlow to monitor the network interface and collect IP network traffic as it enters and exits the interface. A network administrator can determine things such as the source and the destination traffic and class of service by analyzing the data provided. A typical NetFlow architecture has three main components Flow exporter- accumulates the network traffic and exports the flow towards flow collectors, Flow collector- receives and preprocesses the data and finally stores the data, Analysis application- Segregates the flowing data and profiles it on the basis of need. The compressed and preprocessed version of actual network packets are included in NetFlow data.

2.3 Other Data Sets

The DARPA (Defense Advanced Research Project Agency) has two datasets that are invaluable for cyber security researchers. The DARPA 1998 and 1999 dataset was developed by Cyber Systems and Technology group of the Massachusetts Institute of Technology Lincoln Laboratory (MIT/LL). KDD 1999 is another famous data set that is predominantly used by cyber security researchers. Another prominent data set involving SCADA protocol was generated by the Mississippi State University's critical Infrastructure protection center [1]. This dataset will be analyzed in the later sections to evaluate the accuracy of ML algorithms on the SCADA protocols. This data set records the data from a simulated gas pipeline and documents 35 distinct attacks on the SCADA system.

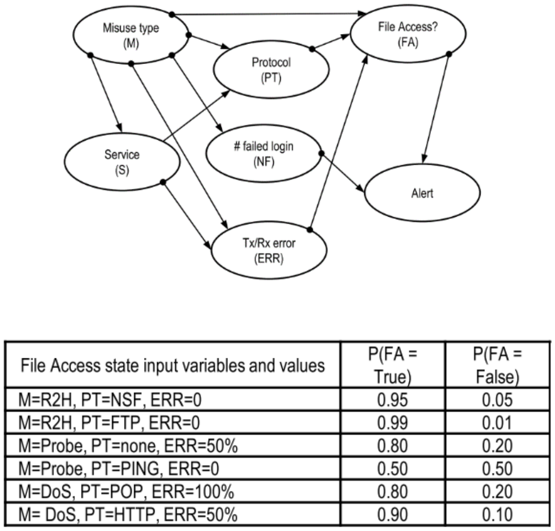
The DARPA 1998 was built on the simulation of network data of TCP/IP, a data of 9 week was collected. 7 weeks of data was used to train the system while the remaining 2 weeks of data was used in the validation of the system [11].Four different types of attacks were defined in this dataset: Denial of service, User to Root, Remote to local and scanning. A similar dataset was prepared again with more simulated attacks, the DARPA 1999 [12]. The KDD 1999 was developed for the KDD cup challenge. This data has 41 detailed attributes and is very similar to the NetFlow data. Tavallae et. al. [13] studied the KDD 1999 data set comprehensively. The statistical analysis revealed that the dataset had a huge number of redundant entries. 78%of the training set and 75 % of the testing set was found to be similar. These inherent problems renders the KDD 1999 unusable and hence a new dataset called the NSL-KDD was proposed.

3. MACHINE LEARNING TECHNIQUES FOR CYBER-SECURITY

Few popular ML techniques are described in this section. For each method the application to cyber security have been identified.

3.1 Bayesian Network

The network is developed as a random set of variable and their conditional dependencies via a directed acyclic graph. The nodes representing the child are dependent on the parent nodes and each node maintains the states of the conditional probability form and the random variable. Fig 1 [10] shows the attack signature detection using Bayesian network. Each state is an input to the underlying state with varying state values. The calculated probability tables are calculated and shown in the figure. Bayesian networks can also be used to infer unobserved variables.



File Access state input variables and values	P(FA = True)	P(FA = False)
M=R2H, PT=NSF, ERR=0	0.95	0.05
M=R2H, PT=FTP, ERR=0	0.99	0.01
M=Probe, PT=none, ERR=50%	0.80	0.20
M=Probe, PT=PING, ERR=0	0.50	0.50
M=DoS, PT=POP, ERR=100%	0.80	0.20
M= DoS, PT=HTTP, ERR=50%	0.90	0.10

Fig 1. Attack signature detection using Bayesian network [10]

Bayesian network can be used for anomaly detection and known attack signature and patterns can also be compared with the streaming data for known attacks. Jemili et. al. [14] developed an intrusion detection system using the Bayesian network. The KDD 1999 was used with 9 of its attributes to model the system. A performance of 88% and 89% was achieved in normal and attack scenarios. The model provided detection rate of 99%, 21%,89% and 7% for Probe, scan , DOS and R2L. Since the number of training instances were very low in case of R2L the accuracy of the model suffered substantially.

3.2 Decision trees

The decision tree is very much analogous to a tree. The trees have leaves which represents the various classifications and the branches are the links or features that in-turn provides the path to the classifications. ID3 and C4.5 are few popular algorithms for generating decision trees automatically.

The comparing process of the SNORT rules with the incoming traffic is slow because of the large number of signatures. Kruegel and Toth et al. [15] replaced 150 SNORT rules by using a variant of ID3 algorithm. Their aim was to replace these algorithm by a decision tree model. This would be effective in increasing the speed of processing. Rule clustering was used to replace the Snort rules. This minimizes the number of necessary comparisons. This also allows parallel evaluation hence speeds up the comparison procedure. The clustered rule was applied to DARPA 1999 dataset. The processing speed and efficiency of the develop model was compared with the snort analysis. The model reached a maximum speed up of 105% and the minimum speed up was of 5%. For further experimentation the number of rules replaced was increased from 150 to 1581. Although Toth does not provide any kind of quantitative figures yet the study detected a profound speed up using the decision tree method, secondly the processing time was reduced drastically.

3.3 Clustering

This is an unsupervised learning method where similarity measure is used to group data together. Clustering algorithms can learn from audit data and explicit description of different attack classes by the system administrator is not necessary.

Hendry et. al. [16] demonstrates the application of real-time signature detection using clustering algorithm. The normal and anomalous network traffic was created by a density based clustering scheme known as Simple Logfile Clustering Tool (SLCT). Two clustering schemes are used: Firstly, for detection of normal and attack scenarios, secondly the other scheme can be used to determine the normal traffic in a supervised manner. In this model parameter M is used to define the feature that is contained in the cluster. By setting M parameter to 97%, 98% attack data is detected with a 15% FAR. The signatures are created from the samples of the high density clusters of the model. The KDD dataset was used to validate the generated model. Cluster integrity was used as the performance metrics to improve the accuracy of the model. An accuracy of 70 % to 80 % was achieved for unknown attacks. Considering the unknown nature (new or zero-day) of the attacks this level of accuracy is quite impressive.

3.4 Artificial Neural Networks (ANN)

The ANN behaves mainly like human brain. The neural network has a layer layout. The input from the data actuates the neuron the second layer of the network. Which in turn outputs to the next layer of the hierarchy. This carries on and finally the output is produced by the last layer of the network. The internal network which plays an important part in the neural network are black boxed from the environment and is known as hidden layers. One major drawback of neural network is the huge learning time due to the occurrence of local minima. This approach was prevalent in mid-nineties but due to the advent of support vector machines (SVMs) ANN started to fade away. With the introduction of convolution NN the popularity of neural network is on the rise again. Canady [17] describes an ANN model which makes use of multi category classifier to detect anomalies. RealSecure network monitor was used to generate the data. The attack signatures were built into the system. Around 3000 attacks were simulated by program like Satan and Internet Scanner out of the 10000 recorded attacks. The data preprocessing was performed using nine selected features: ICMP code, ICMP type, source address, destination address, protocol identifier, source port, destination port, raw data length and raw data type. Then the study used the normal and attack data to train the ANN. Canady et. al. report an error rate of 0.058 and 0.070 during training and testing scenarios. Hence, an RMS of 0.070 translates to a normal accuracy of 93% for the testing phase. Here the data is either categorized as normal traffic or as malicious traffic.

3.5 Genetic algorithm and genetic programming

Two of the most popular computation method based on the principle of survival of the fittest is- GA and GP. These algorithms functions on the population of the chromosomes that evolve based on certain operators. The three basic operator used is selection, crossover and mutation. The algorithm is started with a randomly generated population, a fitness value is computed for each individual. This signifies the ability of the each individual to solve the current problem and individuals with higher probability have higher chance of being chosen in the mating pool. Two capable individual will perform the next step called crossover and finally each will undergo mutation. Among the two mutated individual the highest fit chromosome will be rallied over to the next generation.

Abhram et. al. [18] used a simple GP model to develop a classifier for attacks. Three popular GP models were used in this analysis: Linear Genetic Programming (LGP), Gene Expression Programming (GEP) and Multi Expression Programming (MEP). The model made use of different mathematical operators as function sets. The DARPA 1998 data set was used as the prime dataset to validate the generated model. The dataset had 4 main types (U2R, R2L, DoS and probing) of attacks with a total of 24 different attack scenarios. The False alarm rate (FAR) of the above model was as low as 0% to 5% depending on the type of attack being investigated.

3.6 Hidden Markov Models (HMM)

This is a statistical Markov model with a set of states which are interconnected using transition probabilities that determines the topology of the model. The system is assumed to be a Markov process with unobserved parameters. This model provides a forward- backward correlation which can be used to determine the hidden parameters from the observable parameters. Since the probability distribution in each state is different the system can change states overtime and is capable of representing non-stationary sequences.

Joshi et. al. [19] made use of HMM to develop an intrusion detection system. Five definite states are used each having six observation symbol per state. The interconnection between the states are developed in such a way that any state can transition into any different state. To estimate the HMM parameters the Baum-Welch method can be used. For the validation of the model the KDD 1999 dataset was used. Out of the extensive 41 features of the datasets 5 was chosen for the analysis. The positive detection rate of the model amounted to 79% with a false positive rate of 21%. If more than 5 features are used in the analysis the accuracy of the model might improve but no quantitative analysis was performed by the authors to support this improvement claim.

3.7 Inductive Learning

The inference of certain information from a dataset is known as deduction. On the other hand the other approach of moving from specific observation to develop theories and patterns is known as inductive learning. These are the two primary methods used for the inference of information from the data. Inductive analysis develops some general patterns and which are used to develop some hypothetical conclusions.

Fan et. al. [20] developed an artificial anomaly generator to generate random events and anomalous traffic. Two prime approaches of distribution based anomaly generation and the filtered artificial anomalies was used to generate these random anomaly. This generated data was randomly fused with the DARPA 1998 dataset. This data was used by Fan et. al. to study the performance of the developed inductive learning model. The model showed a successful detection rate of 94 and a low FAR of 2% was achieved. This study enumerated the correct methodology to develop the dataset that can be used for anomaly detection and showed the application of inductive learning model on the developed dataset.

4. ML RECOMMENDATIONS FOR ANOMALY DETECTION

Machine learning is used in cyber-security in three main areas: IDS, Anomaly detection module and misuse detection. Anomaly detection is specifically aimed at segregating abnormal traffic from normal one while misuse detection classifies attack signature comparing it with known ones.

Clustering algorithm (Density based like DBSCAN) works the best with anomaly detection. Apart from the high processing speed clustering algorithms are easy to implement and the parameters to configure are also less in number. SVM also performs considerably well for anomaly detection. For misuse detection the classifiers has to have the capability to generate signatures. Branch feature in a decision tree or chromosomes in genetic algorithm generates signatures that are apt for such task. Hence algorithms like ANN and SVMs which has hidden nodes are not well suited.

5. EVALUATION OF ML ALGORITHMS ON MODBUS DATA

The main aim of this evaluation is to test the applicability of certain ML algorithms to detect cyber-attacks on MODBUS data. Tenfold cross validation was used to develop the ML models. This analysis was performed in Weka [6]. In 10 fold cross validation Weka produces 10 different models for the data set provided. Then the weighted average of these models are calculated which is showed as the final result. The data set used was labeled telemetry data from gas pipeline developed by the Critical Infrastructure Protection Centre of Mississippi state university [1].

Few standard classifier was considered for the evaluation. The methods used were:-

1. **Naïve Bayes**- Bayes' theorem based probabilistic classifier.
2. **Random Forest**-A Ml based on decision tree algorithms.
3. **OneR**- Each feature of the rule set is evaluated and finally the optimum or the best one is chosen.
4. **J48**- A basic implementation of C4.5 decson tree algorithm

5.1 Information about the Dataset

The dataset used was in Weka minable arff format. It had 20 total attributes. The Table I below enlists all the features present in the dataset.

Table I

Features	
address	reset rate
control scheme	command response
function	deadband
pump	time
length	cycle time
solenoid	binary result
setpoint	rate
pressure measurement	categorized result
gain	system mode
crc rate	specific result

Morris et. al.[1] Gives a comprehensive overview about each features of the dataset and why each aspect is important from the perspective of cyber security and intrusion detection. In this dataset a total of 35 attack was performed. These attack can be broadly classified into 7 categories: Naïve Malicious Response Injection (NMRI), Complex Malicious Response Injection (CMRI), Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI), Malicious Function Code Injection (MFCI), Denial of Service (DoS) and Reconnaissance. The Final class in the arff data set has these 7 attack catagories along with normal traffic data. 97019 Instances were recorded in the dataset. The Distribution of the final class is enlisted below in the Table II.

Table II

Class Label	Count
Normal	61156
Naïve Malicious Response Injection (NMRI)	2763
Complex Malicious Response Injection (CMRI)	15466
Malicious State Command Injection (MSCI)	782
Malicious Parameter Command Injection (MPCI)	7637
Malicious Function Code Injection (MFCI)	573
Denial of Service (DoS)	1837
Reconnaissance	6805

.

5.2 Accuracy and ROC curve for ML Algorithm Evaluation

Beaver et. al. [21] have already used the ICS dataset for ML algorithm analysis. But ROC curve was not plotted for any algorithm and hence it is very hard to make out the overall performance of the algorithms. The receiver operating characteristics (ROC) curve is the plot of false positive rate (FAR) in the x-axis versus the plot of test sensitivity in the y-axis. The area under the curve of the ROC is an important parameter. It is used to measure the sensitivity and the specificity. Where the sensitivity is the number of true positive decisions and the specificity is known as the number of true negative decisions. Hence the area under the ROC curve is the combined measure of the sensitivity and specificity. As the area under the ROC curve is the measure of the overall performance of any test hence this parameter can be used to assess the overall performance of the ML algorithms used in the classification of the MODBUS data. Hence an AUC analysis of the ROC curve for different ML algorithm will reveal the classification performance of the algorithms.

The Weka Knowledge flow model for the current analysis was developed. It is shown in Figure 2. The Roc curve generated for the four ML algorithms are show in Figure 5. From the ROC curve it is evident that j48 algorithm produces the most optimized results in general overall classification for the power system dataset. The AUC of the four algorithms is shown in the table below.

Table III

Algorithm	Area under the Curve (AUC)	Precision	Recall
OneR	0.887	0.862	0.894
Naïve Bayes	0.967	0.947	0.936
Random Forest	0.989	0.988	0.988
J 48	0.995	0.992	0.992

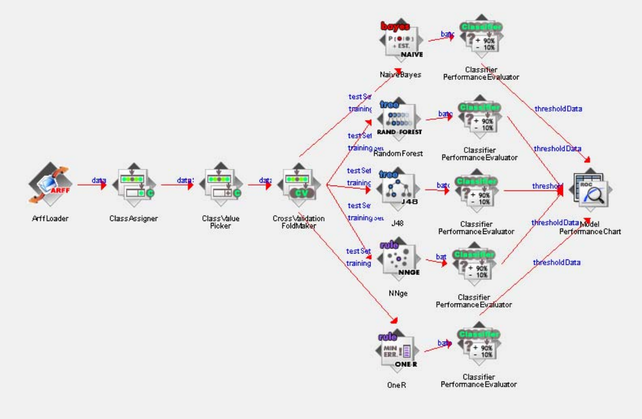


Fig 2. Weka Knowledge flow model for generating the ROC curves

This analysis was done as a binary classification problem. A different approach of multiclass classification can also be used. Secondly in the calculation involving the AUC, Precision and Recall shown in Table III, the weighted average of all 8 class is considered. Each class can be separately analyzed which will provide information about the model's capability to classify each type of attack from each other and from the normal traffic. The results given in Table III shows the model's capability to classify the traffic as whole.

From Fig. 3 it is evident that the J48 performs the best in the overall classification as the area under the curve value for the ROC curves is closer to 1. The ROC graph generated from weka is shown in the appendix as figure 5. In industrial control systems the execution efficiency of the machine learning intrusion detection system being used is of utmost importance. Hence the training must be optimal so that newly streamed data can be trained within a reasonable amount of time and hence the algorithm can still maintain its real-time data monitoring. Therefore, when an algorithm is chosen an optimal accuracy and training time pair is often preferred for each domain of the industrial control system. A python script was used to measure the training time of the algorithm during the K-fold validation. The measured training time is time taken to validate all folds hence in practical scenario the training time can be estimated to be 1/Kth of the plotted time in millisecond as depicted in the figure 4. The machine used for the training has Intel i7 6700HQ as primary processor with Nvidia GTX 1060 dedicated graphic support. Nvidia GTX 1060 has 1256 Cuda cores which greatly parallelizes the training performance.

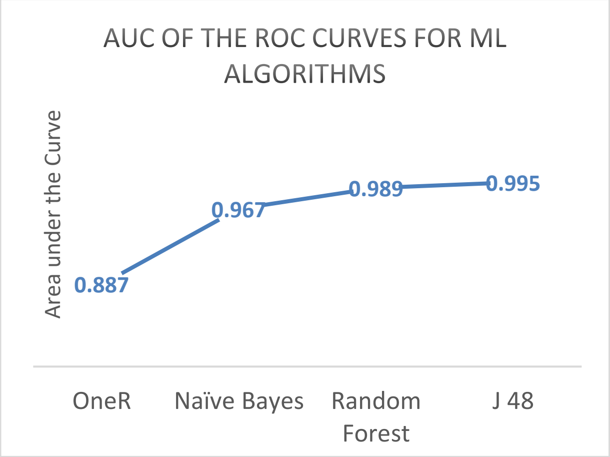


Fig. 3 Area under the curve for all 4 ML algorithm

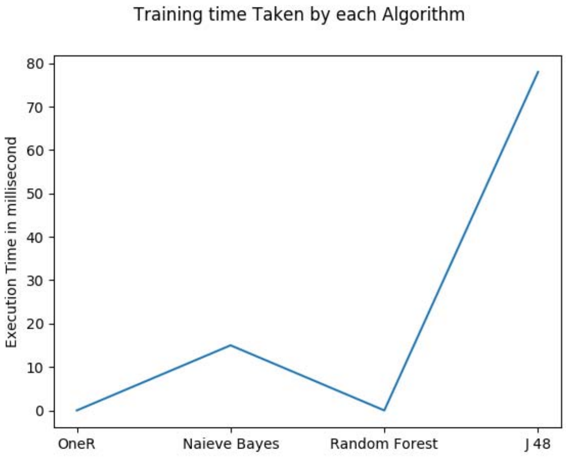


Fig. 4 Training time taken by each algorithm

Hence in this 9 bus IEEE system even though J48 might outperform Random forest in terms of accuracy a little compromise on the accuracy can yield better real-time performance when the algorithms are implemented as a core part of the intrusion detection system.

6. CONCLUSION

In this paper an elaborate survey was performed to enlist few popular datasets then few ML algorithms were discussed along with their application in cyber-security. Finally few recommendations were made regarding the choice of ML. In the later part of the paper a brief analysis was performed with an ICS data set and performance of a few ML algorithm was evaluated. Although J48 algorithm performs better than other algorithms in the scope of analysis, more analysis needs to be performed to ascertain the performance of the algorithms because the performance of algorithms tends to skewed depending upon the dataset on which it is being applied on. Secondly, Random forest might be more suitable as a core IDS algorithm for its optimal real-time performance in the current scenario being considered.

7. ACKNOWLEDGEMENT

I would like to thank Dr. Thomas Morris and his students at the Mississippi State University's Critical Infrastructure Protection Centre for developing the dataset that made this analysis possible.

8. REFERENCES

[1] Morris, T. H., Thornton, Z., & Turnipseed, I. (n.d.). Industrial Control System Simulation and Data Logging for Intrusion Detection System Research.

[2] Nguyen, T. T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. Communications Surveys & Tutorials, IEEE, 10(4), 56–76. http://doi.org/10.1109/SURV.2008.080406

[3] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," IEEE Communications Surveys & Tutorials, 12(3), 2010, pp. 343–356

[4] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. IEEE Communications Surveys and Tutorials, 15(4), 2070–2090. http://doi.org/10.1109/SURV.2013.030713.00020

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & security 28, no. 1, 2009, pp. 18–28

[6] M. Hall, E. Frank, J. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," ACM SIGKDD Explorations Newsletter, 11 (1), 2009, pp. 10–18

[7] R. Core Team, "R Language Definition," 2000

[8] M. Graczyk, T. Lasota, and B. Trawinski, "Comparative analysis of premises valuation models using KEEL, RapidMiner, and WEKA," Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. Springer Berlin Heidelberg, 2009, pp. 800–812

[9] Fowler, C. A., & Hammel, R. J. (2014). Converting PCAPs into Weka mineable data. 2014 IEEE/ACIS 15th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2014 - Proceedings. http://doi.org/10.1109/SNPD.2014.6888681

[10] Buczak, A., & Guven, E. (2015). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, (1), 1–1. http://doi.org/10.1109/COMST.2015.2494502

[11] R. Lippmann, J. Haines, D. Fried, J. Korba, and K. Das, "The 1999 DARPA offline intrusion detection evaluation," Computer Networks, 34, 2000, pp. 579–595

[12] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman," Evaluating Intrusion Detection Systems: the 1998 DARPA Offline Intrusion Detection Evaluation," Proceedings of the DARPA Information Survivability Conference and Exposition, Institute of Electrical and Electronics Engineers (IEEE) Computer Society Press, Los Alamitos, CA, 2000, pp. 12–26

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD Cup 1999 data set," Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, 2009

[14] F. Jemili, M. Zaghdoud, and A. Ben, "A framework for an adaptive intrusion detection system using Bayesian network," Intelligence and Security Informatics, IEEE, 2007

[15] C. Kruegel and T. Toth, "Using decision trees to improve signature- based intrusion detection," Proceedings of the 6th International Workshop on the Recent Advances in Intrusion Detection, West Lafayette, IN, 2003, pp. 173–191

[16] R. Hendry and S. J. Yang, "Intrusion signature creation via clustering anomalies," SPIE Defense and Security Symposium, International Society for Optics and Photonics, 2008

[17] J. Cannady, "Artificial neural networks for misuse detection," Proceedings of the 1998 National Information Systems Security Conference, Arlington, VA, 1998, pp. 443–456

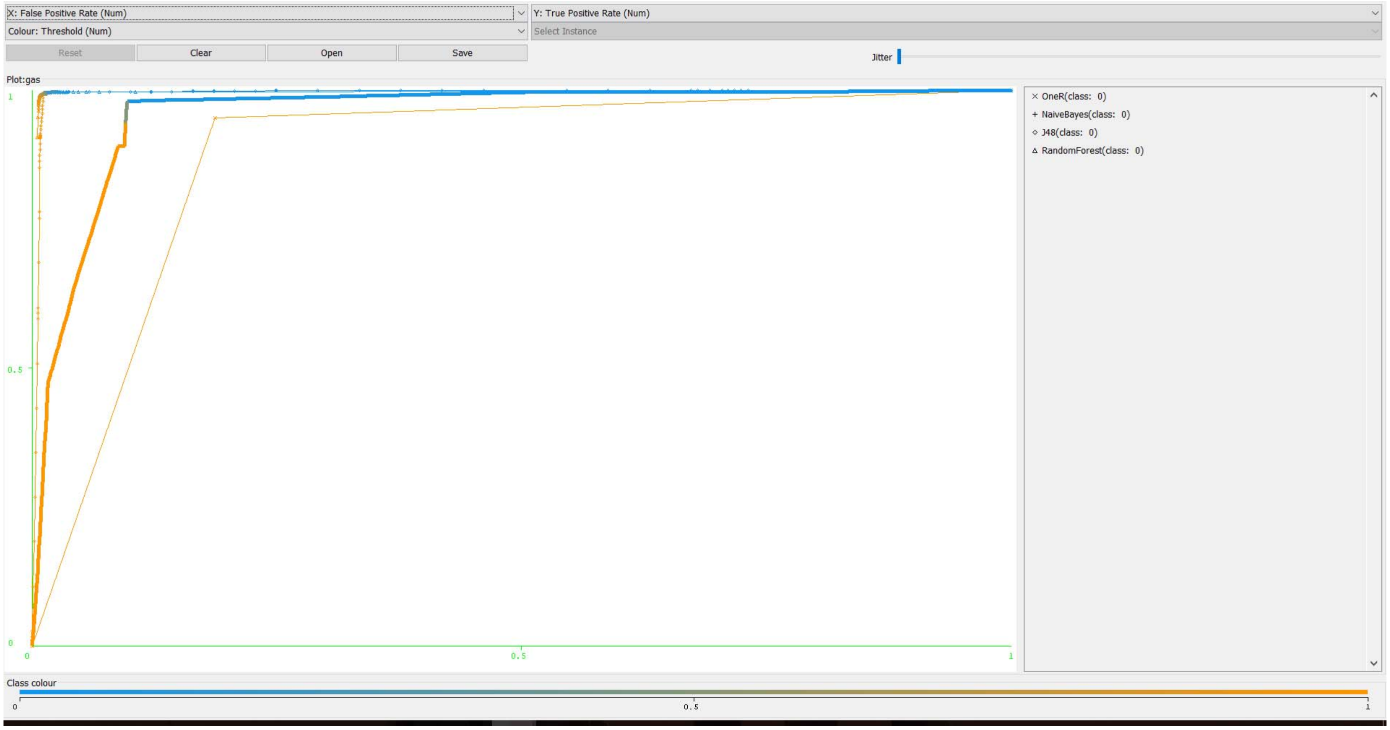
[18] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs," International Journal of Networks Security, 4 (3), 2007, pp. 328–339

[19] S. S. Joshi and V. V. Phoha, "Investigating hidden Markov models capabilities in anomaly detection," Proceedings of the 43rd Annual Southeast Regional Conference, Vol. 1, ACM, 2005, pp. 98–103

[20] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," Knowledge and Information Systems, 6 (5), 2004, pp. 507–527

[21] Beaver, J. M., Borges-Hink, R. C., & Buckner, M. a. (2013). An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications. *2013 12th International Conference on Machine Learning and Applications*, 2, 54–59. http://doi.org/10.1109/ICMLA.2013.105

9. APPENDIX



X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Figure 5. ROC Curves generated from the dataset using four ML algorithm