Towards a Novel Intrusion Detection Architecture using Artificial Intelligence

Salam Khanji
College of Technological Innovation Zayed University
Abu Dhabi, UAE
M80006416@zu.ac.ae

Asad Khattak
College of Technological Innovation Zayed University
Abu Dhabi, UAE
Asad.Khattak@zu.ac.ae

ABSTRACT

Artificial intelligence (AI) is a transformative technology for potential replacement of human tasks and activities within industrial, social, intellectual, and digital applications. Network intrusion detection is crucial to identify cyber-attacks in critical infrastructures where a dynamic collection and analysis of network traffic can be conducted using AI. In this research paper we develop a novel intrusion detection architecture to mitigate malicious traffic passing through cyber infrastructure of an organization. We propose to design scenarios based on AI for intelligent self-protection or alert system that will facilitate countering actual cyber-attacks. The system will utilize machine learning algorithm - Random Forest - to offer more flexibility to discover new attacks and to ensure training the system to predict them in the future. Moreover, we design spam filtering program on python to detect spam emails as per email is one of the main attacking vectors that threatens the security of critical infrastructures.

CCS Concepts

• Security and privacy → Intrusion/anomaly detection and malware mitigation →Intrusion detection systems

Keywords

Artificial Intelligence; Network Security; Intrusion Detection; Cyber-Attacks.

1. INTRODUCTION

With the recent advancement of technologies and emergence of concepts like Cloud Computing, Social Networking, and Big Data; an enormous amount of data is transmitted over a network and intrusion detection systems (IDS) must dynamically collect and analyze this data to eliminate potential threats and cyber-attacks. However; traditional intrusion detection systems need tremendous human interaction to label network traffic either as intrusion or normal one that can be substantially expensive and time consuming.

AI is an emerging technology that consists of several techniques utilized in decision-making process such as machine learning algorithms deployed to identify and respond to detected attacks through classifying features of network traffic to decide if it is either a malicious or not [1]. Other techniques involve detecting anomalies in incoming data to track its behavior and check for any deviation that might be considered as an intrusion. Probabilistic programming is another technique used to create mathematical models of tracked malware in a system so that to perform inference on these models to generate future responses [2]. All mentioned techniques can be combined in hybrid AI architecture for the development of the next generation of innovation and intelligence-driven applications in wide ranges as in neuroscience, healthcare, and cybersecurity. AI technologies are predicted to leverage innovation and the economic growth by reaching overall human ability by 2075 [3]

ICSIE 2020, November 11–13, 2020, Cairo, Egypt

ACM ISBN 978-1-4503-7721-8/20/04…$15.00
DOI: https://doi.org/10.1145/3436829.3436842

In this research paper we propose a novel self-protection architecture based on machine learning to detect anomalies in a network. The proposed work investigates several machine learning algorithms to compare their accuracy in predicting network attacks such as Naïve Bayes, Decision Table, Random Forest, and Adaboost. The objective is to build a flexible protection mechanism where if any new type of attack occurs in the future, the system can be trained to predict its presence in the network traffic. Moreover, we design a spam filtering program to identify and prevent malicious emails from passing the network. The rest of the paper is organized as follows: Section 2 briefly details models and techniques which integrate IDS and AI in the literature. Section 3 demonstrates our proposed model and identifies its settings and specifications. Results and findings are discussed in Section 4 while we conclude in Section 5.

2. BACKGROUND

Security is imperative to maintain in computer networks where IDS offers protection mechanisms to proactively ensure network integrity, confidentiality, and availability. IDS can be categorized to two main groups: network based and host-based IDS [4] where network-based monitors activities of inspected packets through utilizing anti-thread software to control incoming and outgoing threads. Meanwhile, the host-based IDS is integrated into the computer framework to detect abnormalities and to protect the framework's data. In [5] authors proposed an AI based intrusion detection system using deep neural network where it consisted of 4 hidden layers that used non-linear ReUL as the activation function and 100 hidden units to enhance their model's performance. They were able to achieve 99% of accuracy in all scenarios proposed. Another technique is proposed in [6] where authors deployed Recursive Feature Addition (RFA) in network-based intrusion detection system and bigram techniques to encode the payload string features to enhance the feature selection process. The technique suggested an accurate evaluation metric to measure both detection rate and false alarm for different systems.

As attacks on computer networks are becoming more sophisticated and vicious, it is imperative to have adaptive and dynamic intrusion detection systems. In [7] authors addressed intrusion detection adaptability where they proposed a dynamic single-layer solution to detect novel attacks according to new trends of data patterns updated by human expert through clustering and extreme learning machines. However, in [8] authors proposed to use a multi-level adaptive coupled method through utilizing the white list technology and machine learning. The white list was used to filter communication behaviors on the first level, and then machine learning was used to anomaly detect the abnormal communication behaviors on the second level. Their work resulted in a better intrusion detection compared to other single detection algorithms. Whereas in [9] authors proposed an adaptive system based on fuzzy rough set theory to achieve optimal attribute subset of network connection records. Greedy algorithm-based global optimal Gaussian mixture model (GMM) clustering method was used to extract the intrinsic structure of network instances to achieve stable normal intrusion pattern libraries. This resulted in significant improvements in detection accuracies and low false alarms on known and unknow attacks.

3. PROPOSED WORKING MODEL

Our working model consists of two main case scenarios: network intrusion detection against cyber-attacks, and spam filtering program. In the below subsections we will describe each scenario’s specifications and settings.

Table 1. KDD99 Dataset Attacks Description

Categories	Attacks	Instances
DoS	SMURF	2807886
	NEPTUNE	1072017
	Back	2203
	POD	264
	Teardrop	979
U2R	Buffer overflow	30
	Load module	9
	PERL	3
	Rootkit	10
R2L	FTP write	8
	Guess pswd	53
	IMAP	12
	MultiHop	7
	PHF	4
	SPY	2
	Warez client	1020
	Warez master	20
PROBE	IPSWEEP	12481
	NMAP	2316
	PORTSWEEP	10413
	SATAN	15892
Normal		972781

3.1 Network Intrusion Detection

3.1.1 Dataset

It is imperative to build a reliable network where the rapid development of digital technologies imposes new security threats and challenges. The Denial of Service (DoS) attack is one the most popular risk that threatens computer networks and affects significantly system availability. A predefined dataset is used – KDD99 - which contains several types of attacks such as DoS, U2R, R2L, PROBE, and normal (no attack). A total of 21 attacks under main attack categories mentioned before and as illustrated in Table 1.

The dataset contains a total of 41 attributes to decide if a traffic is malicious or not. The dataset is pre-processed to remove any redundancy or missing values then to generate a CSV file to be fed to the system that is built using R programming language and WEKA; an open source Java platform for classifying and clustering.

3.1.2 Implementation

WEKA is utilized as a mining tool to compare between several machine learning algorithms in terms of accuracy using cross validation (10 fold) technique and the percentage split (70%) technique as well. The selected machine learning algorithms are: Naïve Bayes, Decision Table, Random Forest, and Adaboost. Comparison techniques deployed in WEKA estimates the correctly classified instances for each machine learning algorithms and results are demonstrated in Table 2.

Table 2. Machine Learning Algorithms Accuracy

	Cross Validation (10 Fold)		**Percentage Split (70%)**	
	Correctly classified instances	Incorrectly classified instances	Correctly classified instances	Incorrectly classified instances
Naïve Bayes	58866 (76.16%)	18245 (23.84%)	17987 (77.57%)	5200 (22.43%)
Decision Table	76141 (98.51%)	1150 (1.49%)	22868 (98.51%)	319 (1.36%)
Random Forest	76829 (99.4%)	462 (0.6%)	23040 (99.37%)	147 (0.63%)
Adaboost	68113 (88.13%)	9178 (11.87%)	20441 (88.16%)	2746 (11.84%)

As per the results tabulated in Table 2, Random Forest scored 99.4% of correctly classified instances, consequently; Random Forest was elected to be implemented in building the network intrusion detection system.

3.1.3 Random Forest

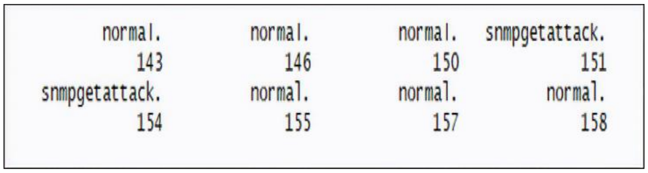
Random Forest is an ensemble supervised machine learning algorithm where its classification performance is better than other single classifier models and can handle both binary classification problems and multiclassification problems [11]. It uses random sampling combined with replacement to form multiple decision trees where the final result is achieved by voting. The process of RF is as follows.(1)Using randomly sampling with replacement to extract samples from dataset and acquire a training subset.(2)For the training subset, *m* features are randomly extracted from the feature set without replacement as the root for splitting each node in the decision tree. From the root node, a complete decision tree is created from top to bottom.(3)The *k* decision trees are generated by executing steps (1) and (2) repeatedly K times. RF classifier is attained by combining these decision trees where the result of classification is nominated by these decision trees [12].

3.1.4 R Language

Since Random Forest classifies instances based on ensemble learning through building multiple decision trees to form the output function, this would increase the computational time significantly. Hence, the InfoGain attribute is used as a feature selection method in WEKA that measures how each feature contributes in decreasing the overall entropy so that to lessen the computational time of the Random Forest algorithm. However, this would decrease the accuracy rate of our proposed system, nevertheless; it would make intrusion detection more efficient as per it will consume less time in detecting possible attacks. 5 attributes have been elected when using the InfoGain attribute method out of the 41 attributes in the dataset which are:

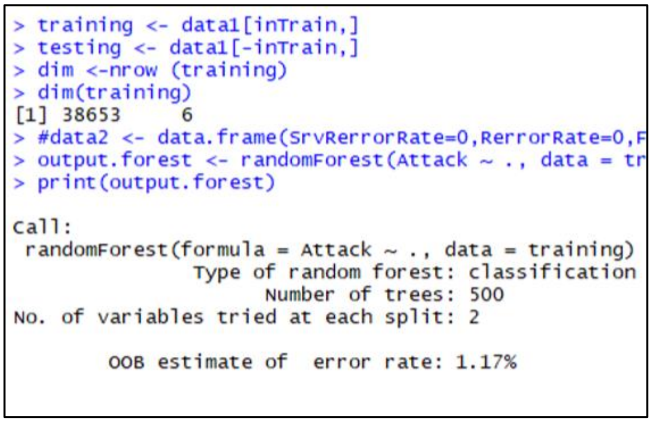
- *SrcBytes:* number of data bytes from source to destination.
- *DstBytes:* number of data bytes from destination to source.
- *DstHostSambeSrvRate:* destination host same server rate.
- *Count:* number of connections to the same host for the current connection in the past two seconds.
- *DstHostDiffSrvRate:* destination host different server rate.

Then, R language is used to build the intrusion detection system that deploys both the Random Forest machine learning algorithm and the InfoGain attribute method on the processed KDD99 dataset. Figure 1 shows some of the predicted output of the test data when running the R source code. The program will predict the error rate, type of network traffic if it is malicious or normal, and it will continuously adapt to capture new types of attacks in the future.



```
      normal.        normal.        normal.  snmpgetattack.
          143            146            150             151
snmpgetattack.       normal.        normal.         normal.
          154            155            157             158
```

Figure 1. Predicted output of the test data



```
> training <- data1[inTrain,]
> testing <- data1[-inTrain,]
> dim <-nrow (training)
> dim(training)
[1] 38653     6
> #data2 <- data.frame(SrvRerrorRate=0,RerrorRate=0,F
> output.forest <- randomForest(Attack ~ ., data = tr
> print(output.forest)

Call:
 randomForest(formula = Attack ~ ., data = training)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 1.17%
```

Figure 2. OOB Estimate of Error rate

The OOB (Out of Bag/Prediction error) Estimate of Error of Random Forest when applied on the dataset (half of it as training data and the other half as test data) scored 1.46% which is mapped to 98.54% of accuracy when detecting intrusions as demonstrated in Figure 2 and Figure 3. Consequently, we can predict if the connection is normal or malicious and we can also identify each attack type such as Neptune, Saturn, rootkit, or any other type.

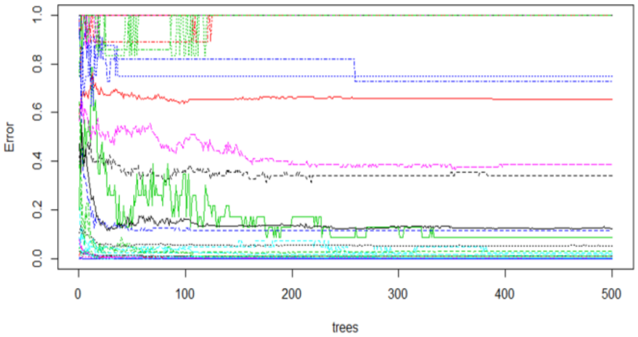


Figure 3. Random Forest output graph

3.1.5 Implementation

A spam classifier is created using Python and Anaconda (Jupyter Notebook) to run it. The program utilized the Bayes' Theorem to identify the occurrence of words collected in CSV file where they have been classified and refined as spam or non-spam. The dataset of emails is featured by 1 for spam email, and 0 for non-spam. To test our model we should split the data into train dataset and test dataset. We used 75% of the dataset as train dataset and the rest as test dataset. Selection of this 75% of the data is uniformly random. The program workflow is:

3.1.5.1 Generate Dictionary

In this step we create a dictionary of words that are used as features by the algorithm to decide if any given email is a spam or not. We choose the first 2500 most frequent words counting all emails in the four groups (spam and non-spam test emails, spam and non-spam train emails). This can be done through counting the number of occurrences of each word in all emails and then select the most frequent 2500 words. Results are tabulated in a CSV file (dictionary) that classifies words where some are considered spam and others are non-spam according from which of the four groups they have been selected.

3.1.5.2 Generate Features

In this step we extract features from both train and test emails so the result structure is prepared as an input to the Na ïve Bayes algorithm utilized in the prediction model. We only need to count number of occurrences of each word collected in the dictionary in all emails from all the four groups mentioned before. The results form labels where 0 marks non-spam emails and 1 marks spam emails.

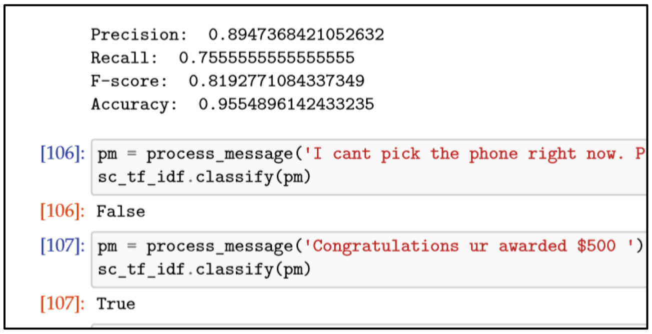
3.1.5.3 Generate and Test the Model

The program is based on the Na ïve Bayes algorithm where data structures prepared from the two steps mentioned (dictionary and features) above are fed into the algorithm to classify if any given email is spam or no-spam. The algorithm is a probabilistic classifier that calculates the probability of an item (email) belongs to a particular category. The formula used is illustrated in Figure 4 where it counts the number of occurrences of a w (one word from the dictionary) in all the c (sum of all occurrences of the dictionary words in ether spam or non-spam emails depending for which one we are estimating the probability). V — is the number of words in the dictionary. This probability will be calculated separately first on spam and then on non-spam emails.

$$\hat{P}(w_i \mid c) = \frac{count(w_i,c)+1}{\sum_{w\in V}(count(w,c)+1)} = \frac{count(w_i,c)+1}{\left(\sum_{w\in V} count(w,c)\right) + |V|}$$

Figure 4. Naive Bayes formula with Laplacian smoothing [14]

In the final step we test our model and the output variable is either true for spam email and false for non-spam email. Moreover we also calculate precision and accuracy rate where the model is predicting with accuracy of 95.5489% as depicted in Figure 5.



```
       Precision:  0.8947368421052632
       Recall:  0.7555555555555555
       F-score:  0.8192771084337349
       Accuracy:  0.9554896142433235

[106]: pm = process_message('I cant pick the phone right now. P
       sc_tf_idf.classify(pm)

[106]: False

[107]: pm = process_message('Congratulations ur awarded $500 ')
       sc_tf_idf.classify(pm)

[107]: True
```

Figure 5. Model Testing Results

4. RESULTS AND DISCUSSION

Our proposed system is designed to identify possible attacks and to detect intrusions in the network traffic dynamically. The system also classifies the incoming emails and filters them to prevent spam emails from passing through the network server. Table 3 summarizes detailed accuracy rates for both scenarios mentioned in Section 3. The first scenario included other attacks subcategories where their detailed accuracy is plotted in Figure 6.

Based on the test results, our proposed system has a significant performance advantages to detect intrusions and to further adapt to detect future attacks. The system utilized the InfoGain feature selection method to lessen the computational overhead so that to offer a more flexible intrusion detection process.

Table 3. Accuracy Results for the Proposed System

Scenario	Accuracy	Precision	Recall	F-Measure
Intrusion Detection	0.994	0.967	0.878	0.967
Spam Filter	0.955	0.894	0.755	0.819

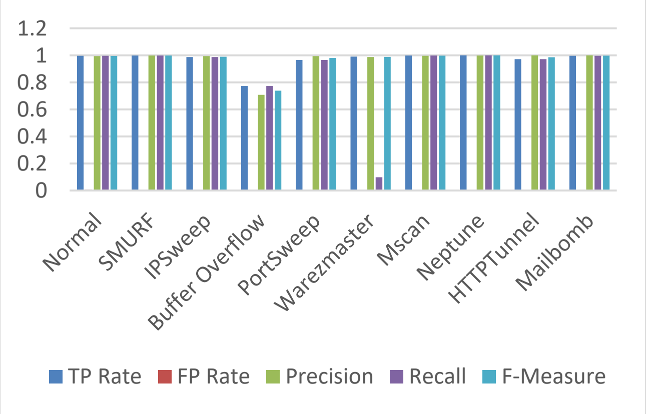


Figure 6. Intrusion Detection – Detailed Accuracy by Class

However, the dataset used in both scenario cases would not necessarily represent a real world scenario of scale and a variety of applications. Nevertheless, the performance on the test dataset is a good indicator for general system's performance.

Applying machine learning techniques in intrusion detection system would harness the system security by efficiently detecting anomalies or attacks. However, there are several challenges that must be tackled when considering machine learning or AI in IDS. Some of these challenges are the outlier detection, high cost of errors, semantic gap, and difficulties in evaluation [13]. The outlier detection problem is when it is difficult to identify a normal profile of a network when it varies in different sessions and different applications. Consequently, this would cause a misclassifications where either the system does not detect an attack as per it resembles a normal network traffic, or the system detects a false attack as per the traffic is generated from a different application with different network traffic pattern that the system has not encountered before.

Another challenge is high cost of errors that can occur in IDS and can be more serious when applied in spam filter program for. For instance, if a false positive occurs this might cause an expensive analysis time lost. Meanwhile, if a false negative occurs it might cause a significant damage for the infrastructure. The semantic challenge is another problem that would need a high cost expert analyst as per we still need to interpret any detected deviation from the normal profile so that to be able to link the deviation to the right action.

It is crucial to evaluate any dynamic IDS so that to avoid the high cost errors challenge mentioned before. However, it is difficult to evaluate the system accurately due to the lack of datasets to assess the performance of the system in real world scenarios since real datasets are not usually available because of its sensitivity and the simulated datasets are not necessary accurate. Consequently, it is difficult to design a sound evaluation scheme for machine learning based IDS because it is hard to predict accurately how the attack will behave in the future.

5. CONCLUSION

With the increasing number of malicious programs on the Internet, the continuous development of cyber-attacks, and the expansion of intelligent devices; it has become a consensus to introduce AI into intrusion detection systems so that to automatically improve systems' performance, detection rate, and to reduce false positive rate as well. Machine learning is an application of AI where it provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In this research work we proposed a novel machine learning-based intrusion detection system that harnesses crucial cybersecurity network of critical infrastructure. Our model compared several machine learning algorithms in terms of accuracy rate and later deployed Random Forest that was 99.4% accurate of correctly classified instances to be utilized as a classifying algorithm to detect intrusions in network traffic. The OOB Estimate of Error was found to be 1.46% when the half of dataset was used in training and the other half in testing. For each of the testing dataset we can predict if the connection setup is normal or malicious and even pin point to particular type of attack as Neptune, smurf, Saturn, teardrop, rootkit, or any other type. Hence, we can be 98.54% sure of the system's prediction value when deploying the algorithm running on R Language.

Moreover, we designed a spam filter program to classify emails and to prevent spam emails from passing through the network. The program is developed on Python where it pre-processes emails to remove any redundancy, interpunctions, or special characters, and then to classify emails through Na ïve Bayes algorithm to spam or non-spam emails according to the generated dictionary of words. The program's accuracy rate and precision rate were found to be 95.5% and 89.4% respectively. The proposed work can be utilized as a novel self-protection architecture to allow critical network infrastructure to be able to protect against cyber-attacks by means of the dynamic reaction against malicious traffic passing through the cyber infrastructure of an organization.

6. ACKNOWLEDGMENTS

This research was supported by Zayed University Provost Research Fellowship Award R18114.

7. REFERENCES

[1] Rajan, H. M., Dharani, S., & Sagar, V. (2017). Artificial Intelligence in Cyber Security–an Investigation. *International Research Journal of Computer Science (IRJCS)* ISSN, 2393-9842.

[2] Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., ... & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1), 973-978.

[3] M üller, V. C., & Bostrom, N. (2016). Future progress in artificial intelligence: A survey of expert opinion. In *Fundamental issues of artificial intelligence* (pp. 555-572). Springer, Cham.

[4] Samrin, R., & Vasumathi, D. (2017, December). Review on anomaly based network intrusion detection system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)* (pp. 141-147). IEEE.

[5] Kim, J., Shin, N., Jo, S. Y., & Kim, S. H. (2017, February). Method of intrusion detection using deep neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 313-316). IEEE.

[6] Hamed, T., Dara, R., & Kremer, S. C. (2018). Network intrusion detection system based on recursive feature addition and bigram technique. *Computers & Security*, 73, 137-155.

[7] Roshan, S., Miche, Y., Akusok, A., & Lendasse, A. (2018). Adaptive and online network intrusion detection system using clustering and extreme learning machines. *Journal of the Franklin Institute*, 355(4), 1752-1779.

[8] Chen, W., Liu, T., Tang, Y., & Xu, D. (2019). Multi-level adaptive coupled method for industrial control networks safety based on machine learning. *Safety Science*, 120, 268-275.

[9] Liu, J., Zhang, W., Tang, Z., Xie, Y., Ma, T., Zhang, J., ... & Niyoyita, J. P. (2020). Adaptive intrusion detection via GA-GOGMM-based pattern learning with fuzzy rough set-based attribute selection. *Expert Systems with Applications*, 139, 112845.

[10] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information retrieval*, 6(1), 49-73.

[11] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[12] Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Building an Effective Intrusion Detection System by Using Hybrid Data Optimization *Based on Machine Learning Algorithms. Security and Communication Networks*, 2019.

[13] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy* (pp. 305-316). IEEE.

[14] Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. arXiv preprint arXiv:1410.5329.