THREAT DETECTION IN IoT USING MACHINE LEARNING TECHNIQUES

[1]A. K. Kwansah Ansah, [1]B. O. Antwi, [2]K. K. Bondzie
[1]University of Mines and Technology, P. O. Box 237, Tarkwa, Ghana
[2]Golden Star Resources, P. O. Box 16075, Accra, Ghana

Kwansah Ansah, A. K., Antwi, B. O. and K. K. Bondzie (2022), "Threat Detection in IoT using Machine Learning Techniques"

Abstract

Consumer security and privacy have become key concerns as a result of the increased use of Internet of Things (IoT) devices. Existing security and privacy safeguards are insufficient due to the rapid increase in cyber threats; hence all Internet users are a hacker's target. Machine-learning techniques could be employed to generate precise outputs from large and sophisticated databases, which may then be used to foresee and detect weaknesses in IoT-based systems. Although, several research have been conducted on machine learning regarding IoT security, there are no obvious, direct, or laid-out procedures to follow to detect these threats. This paper seeks to provide a model with sufficient accuracy for detecting intrusion in IoT devices based on the analysis of various machine-learning algorithms. The paper also delves into IoT system flaws and provides possible fixes.

Keywords: IoT Devices, Machine-learning, Security

1 Introduction

Most organizations have evolved from producing just products to building a network of their products (The Internet of things). Computing seeks to ease human lives by taking away everyday simple tasks and automating them or assigning them to machinery. The IoT is no different either. The IoT is a giant network of physical devices that are connected to the internet for collecting and sharing data. IoT has been adopted in a host of areas like healthcare for monitoring calorie count and heart rate, manufacturing for monitoring production flow, finance through the use of smart cash points, and agriculture for monitoring climate conditions. The list is endless.

1.1 The Prevalence of IoT Devices

The number of IoT devices keep increasing exponentially every year. According to a report published by Arne Holst on statista.com in March 2021, the number of IoT devices worldwide is forecast to almost triple from 8.74 billion in 2020 to more than 25.4 billion devices by 2030. In 2020, the highest number of IoT devices was found in China with 3.17 billion devices (Holst, 2021). IoT however comes with a ton of risks mainly security-wise.

1.2 Vulnerabilities in IoT Devices

Security is a major concern in IoT as recent publication and publicity surrounding the IoT has been on cybersecurity. However, security in IoT sense is quite vague as Hugo Fiennes, CEO of Electric Imp pointed out "some kind of accepted framework or standard has to happen" "With IoT devices being so complex, the lack of some kind of checklist leads to security vulnerabilities"- Stacey Higginbotham, Tech Writer and industry analyst. Despite all this, the IoT is here to stay as the future looks limitless. The same, however, can be said for cybercrimes. IoT devices are insecure by nature. They are connected, meaning the bad guys can access them. IoT devices by themselves lack the processing power for basic protection like encryption. They also tend to be highly valuable and inexpensive, making it easy for users to deploy large numbers of them (Elgan, 2020).

1.3 Machine-learning and Threat Detection

Machine learning jumps straight to mind anytime possible solutions to the security vulnerabilities of IoT devices are discussed. Machine learning has already made great strides in the bid to make IoT devices more intelligent. Although the concept of machine learning in IoT security is not new, it is not as straightforward as industries would like it to be because IoT keeps evolving and machine learning algorithms are numerous. However, this paper seeks to establish a better machine-learning algorithm for threat detection depending on the IoT devices or the characteristics of the IoT data in question.

2 Materials, Methods Used

2.1 Methodology

A machine-learning model is created by learning and generalizing from training data, then applying that knowledge to new data to generate predictions and achieve its goal. You won't be able to develop the model if you don't have enough data, and having data isn't enough. Data that is useful must be clean and in good condition. Determining the data requirements and their suitability for the machine-learning project is a huge factor to consider when developing a machine-learning model. In this project, data identification, initial collection, requirements and quality identification was prioritized.

2.2 Collection and Preparation of Data

The accuracy of a model depends greatly on the sort of data that is supplied to the input. Data gathering is, therefore, an essential procedure that cannot be overlooked. Data preparation tasks include structured, unstructured, and semi-structured data gathering, cleansing, aggregation, augmentation, labeling, normalization, and transformation, as well as any additional activities. The dataset used in this project is the NSL-KDD dataset and was downloaded from the website of the University of Brunswick.

2.2.1 The NSL-KDD Dataset

Although identifying the existing original dataset is difficult, the NSL-KDD dataset can nevertheless be used as a useful benchmark dataset for academics to compare different intrusion detection systems as far as the field of IoT is concerned. The attack types in the dataset are grouped into four. Probe, denial-of-service, remote to local and user to root (Bala, 2019). Also, the dataset is pre-divided into testing and training datasets. In Table 1, the 39 attack types available in the dataset have been grouped into four namely, Probe, denial-of-service, remote to local and user to root. The ones in **bold** are the attack types present in the only testing dataset. The dataset has 22 attack types for training and an additional 17 attack types for testing.

Table 1 NSL-KDD Dataset

DoS	Probe	R2L	U2R
Apache2	Ipsweep	Spy	Bufferoverflow
back	**Mscan**	*warezclient*	Loadmodule
Land	nmap	ftp_write	perl
mail bomb	portsweep	guesspasswd	**ps**
Neptune	**saint**	**Httptunnel**	rootkit
pod	satan	imap	**snmpguess**
processtable		Multihop	**sqlattack**
smurf		**named**	**worm**
teardrop		phf	**xterm**
udpstorm		**sendmail**	
		Snmpgetattack	
		warezmaster	
		xlock	
		Xsnoop	

2.2.2 Data Pre-processing and Cleaning

Data pre-processing is the procedure for preparing raw data for use in a machine learning model. Its the first and most important stage in building a machine learning model. Figure 1 represents a snippet of how our data was cleansed. The function def convLabels which returns (df) maps the symbolic names to a label encoded value. This label *n* is used to classify a model. Normal attacks were mapped to 0, dos attacks were mapped to 1, probe attacks mapped to 2, remote to local mapped to 3 and user to root mapped to 4. This is referred to as normalization in the data cleaning process. Data normalization is the process of organizing data such that it seems consistent across all records and fields (Watts, 2020).

The NSL-KDD Dataset came already divided into training and testing datasets. A training set is a collection of data sets that will be used to fit or train a model. The testing sets are the remaining datasets that aren't utilized for training but are instead used to estimate the model's performance to tune it. This provides an unbiased sense of model effectiveness (Khun and Johnson, 2013).

```
def convLabels(df):
    df = df.replace('normal', 0)
    #Denial of Service attacks - Mapped to 1
    df = df.replace('back', 1)
    df = df.replace('land', 1)
    df = df.replace('neptune', 1)
    df = df.replace('pod', 1)
    df = df.replace('smurf', 1)
    df = df.replace('teardrop', 1)
    df = df.replace('mailbomb', 1)
    df = df.replace('processstable', 1)
    df = df.replace('udpstorm', 1)
    df = df.replace('apache2', 1)
    df = df.replace('worm', 1)

    #Probe attacks - Mapped to 2
    df = df.replace('satan', 2)
    df = df.replace('ipsweep', 2)
    df = df.replace('nmap', 2)
    df = df.replace('portsweep',2)
    df = df.replace('mscan', 2)
    df = df.replace('saint', 2)
    df = df.replace('spy', 2)
```

Figure 1 Data Pre-processing

2.3 Classifiers

2.3.1. Naïve Bayes

Naïve Bayes is a probabilistic machine-learning algorithm based on Bayes' theorem. Naïve Bayes Algorithm is a supervised learning algorithm and is used to solve classification problems (Sunil, 2017). For example, an animal may be classified as a duck if it has a wide beak, feathers, walks on two legs and can float on water because there is the probability or likelihood that it is a duck

2.3.2. Linear Regression

Linear regression is one of the simplest and most popular machine learning algorithms. It is a statistical method used for predictive analysis (Brownlee, 2016). The linear regression algorithm displays the linear relationship between the factor and one or more independent variables (y), thus it is called linear regression. Since linear regression shows a linear relationship, it means that it finds how the value of the dependent variable changes based on the value of the independent variable (Brains, 2022). Mathematically, we can express linear regression as:

$$y = a_0 + ax + \varepsilon \quad (1)$$

where

Y = dependent variable (target variable)

X = independent variable (predictor)

a_0 = intersection of lines (provides additional degrees of freedom)

a_1 = linear regression coefficient (scale factor for each input value).

ε = random error

2.3.3. Random Forest

Random forest is a supervised learning technique for solving classification and regression problems (Bonthu, 2021). It creates a forest out of a collection of decision trees and merges them to produce more accurate results (Shah et al., 2020). The basic concept behind random forest is simple but quite potent. The crowd is probably right. It consists of several decision trees. The algorithm (Random Forest) generates results based on the decision tree predictions by taking the average. Increasing the number of trees will increase the precision of the results (Ali et. al., 2012). It is like an election by decision trees where the most common prediction is considered accurate.

2.3.4. K Nearest Neighbors (KNN)

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other (Harrison, 2018). In KNN, the goal is to classify the given new unseen data points and view the given K data points in the training set, which are closest in the input space or feature. Therefore, to find the KNN of a new data point, we must use distance metrics, such as Euclidean distance, L∞ norm, angle, Mahalanobis distance, or Hamming distance (Chomboon *et al*., 2015).

2.4 Model Training

Model training is making constant adjustments to the model's parameters to achieve a desired outcome or output. The model training process begins with selecting a training loss, an optimizer, and continuously calculating the gradient of the loss for the model's parameter to update these parameters using the optimizer. Figure 2 is a flow chart of the procedures used in building the model. The original KDD datasets (training and testing) were downloaded, and data cleaning techniques were applied to the datasets. The various classifiers were employed in training the dataset to predict whether an instance in the dataset is normal or an attack. If it is an attack, it is classified under one of the four attack types present in the dataset.

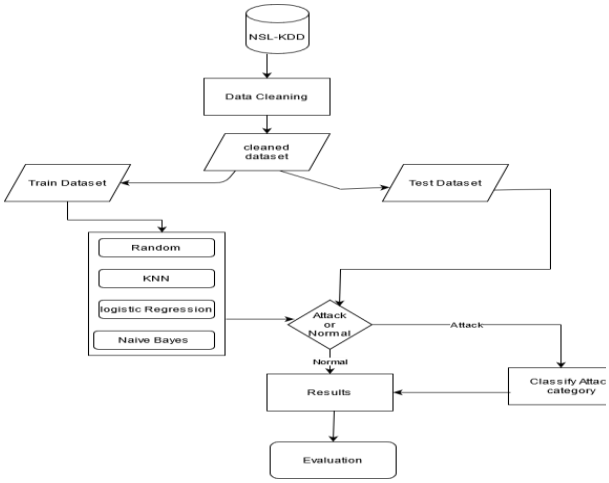

Figure 2 Flow Chart of the Model

3 Results and Discussions

This section presents the experimental results of testing the model on the test set, as well as the performance of the verification that was used in this work for the identification and classification of various attack types in an IoT network. After the model is trained, the inference is needed, that is, the model is tested on the data that has not been seen before. Evaluation is done using the confusion matrix and accuracy curves.

3.1 Confusion Matrix

A confusion matrix is a table that is frequently used to evaluate the performance of a classification model on a set of test data with known true values. The confusion matrix is straightforward and aids in understanding the general performance of a classification model (Gu *et. al.*, 2009). The confusion matrix produced by our model is a 5 X 5 matrix as shown in Figure 3. The accuracy of the model can be calculated by taking the average values along the main diagonal of the matrix.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

where FP = False Positives, FN = False Negatives, TP = True Positives and TN = True Negatives.

Accuracy is also calculated as 1-e where e is the error rate.

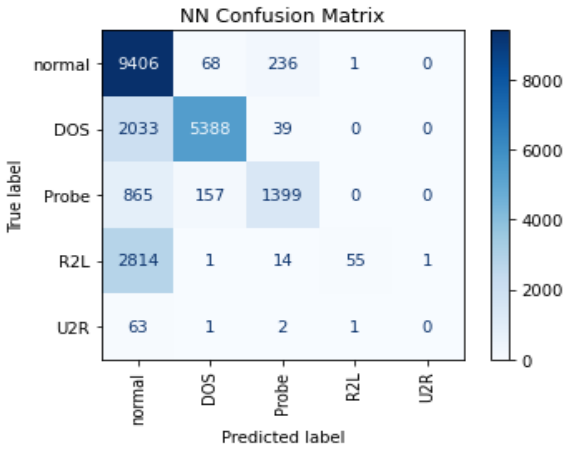

Figure 3 NN Confusion Matrix

The accuracy the matrix gives can be simplified as

$$Accuracy = \frac{\text{True Predictions}}{Total\ Population} \quad (3)$$

$$Accuracy = \frac{16248}{22544} = 0.7207\ or\ 72.07\% \quad (4)$$

3.2 Accuracy Curves

The Accuracy Curves are graphs of the accuracies against the number of iterations used in training (Ge et. al., 2020). From fig 4.2, the accuracy keeps improving steadily as we keep performing the training iterations and finally reach a peak accuracy of 97%. The loss curve is the inverse of the accuracy curve.

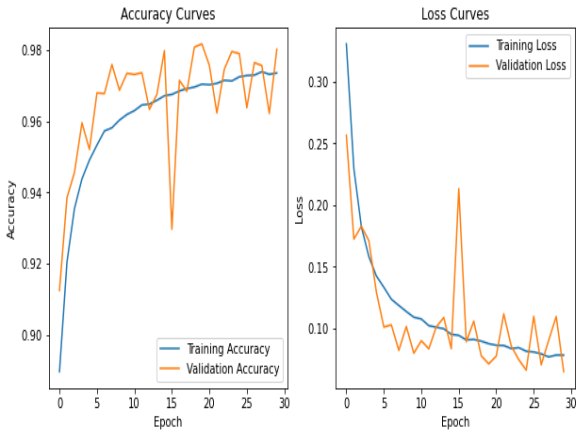

Figure 4 Accuracy Curve

4 Conclusions and Recommendations

The various threats to IoT were analyzed and different machine learning algorithms were deployed to build a single working model which has high accuracy in detecting threats in the IoT.

It is recommended that future works on threat detection in IoT should focus more on obtaining real-time datasets to increase the accuracy of the models.

References

Ali, J., Khan, R., Ahmad, N., and Maqsood, I. (2012). "Random Forests and Decision Trees" *International Journal of Computer Science Issues*(IJCSI). Vol. 9, No. 5, pp. 272 – 278.

Bala, R. (2019). "A Review on KDD Cup99 And NSL-KDD Dataset", *International Journal of Advanced Research in Computer Science*. Vol 10, pp 64-67.

Bonthu, H. (2021), "Understanding Random Forest",*https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.* Accessed: August 14, 2021.

Brains, B., (2022), "Regression Definition" *https://www.investopedia.com/terms/r/regression.asp* . Accessed: August 29, 2022.

Brownlee,J.(2016),"Linear Regression for Machine Learning",*https://machinelearningmastery.com/linear-regression-for-machine-learning* Accessed: August 17,2021.

Chomboon, K., Chujai, P., Teerarassammee, P., Kerdprasop, K. and Kerdprasop, N. (2015). An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm. 280-285. 10.12792/iciae2015.051.

Elgan, M. (2020), "5 IoT Threats to Look Out for in 2021",*https://securityintelligence.com/articles/iot-threats-look-out-2021/*.Accessed: June 9, 2021.

Ge, C., Gu, I., Jakola, A., and Yang, J. (2020). Deep semi-supervised learning for brain tumor classification. BMC Medical Imaging. 20. 10.1186/s12880-020-00485-0.

Gu, Q., Zhu, L., and Cai, Z. (2009). "Evaluation Measures of the Classification Performance of Imbalanced Data Sets", *Communications in Computer and Information Science* Vol. 51, 461-471. 10.1007/978-3-642-04962-0_53.

Harrison, O. (2018), "Machine Learning Basics with the K-Nearest Neighbors Algorithm" *https://towardsdatascience.com/machinelearning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761* . Accessed: August 15, 2021.

Holst, Arne. "Global Annual Mobile Data Usage by Device Type 2025." *Statista*, 26 Mar. 2021, *www.statista.com/statistics/1222706/worldwide-annual-mobile-data-usage-by-device-type/.* Accessed 23 Sept. 2021.

Khun, M. and Johnson, K. (2013), "Applied Predictive Modeling", Springer, Basel, 600pp.

Shah, D., Patel, S., & Kumar, S. (2020), "Heart Disease Prediction using Machine Learning Techniques", *SN Computer Science*, Vol 1. No. 6, pp 1- 6.

Sunil, R. (2017), Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples *https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained* Accessed: August 17,2021.

Watts, S. (2020), "What Is Data Normalization?" *https://www.bmc.com/blogs/data-normalization/*. Accessed: August 2, 2021.

Authors

Engr. Dr Albert Kofi Kwansah Ansah is a professional engineer and holds PhD degree in Computer Science and Technology from the School of Information and Software Engineering of the University of Electronic Science and Technology of China, Chengdu, PR China in 2021, MSc degree in Mobile Computing and Communication from the Computing and Mathematical Science Department of the University of Greenwich, London, UK in 2008 and PG Dip in Networks and Communication from the Westminster College, London, UK in 2006. He is a Lecturer in the Computer Science and Engineering Department at the University of Mines and Technology (UMaT), Ghana. His research interests cover blockchain technology, cryptocurrencies, privacy-preservation and cyberspace security, computer systems and Networks. He is a member of the Institution of Engineering and Technology (IET-GH), International Association of Engineers (IAENG) and the Internet Society (ISOC) UK and Ghana chapters.

Mr. Bright Antwi Owusu holds a BSc in Computer Science and Engineering from the University of Mines and Technology Tarkwa. He is currently serving as a Teaching Assistant with the UMaT School of Railway and Infrastructure Development. His research interests cover the fields of bioinformatics, data analysis, database management and artificial intelligence.

Mr. Kofi Egyin Bondzie holds a BSc in Computer Science and Engineering from the University of Mines and Technology Tarkwa He is currently an IT officer at Golden Star (Wassa) Mines. His research interests cover the fields of cyber-security and data analysis.