IoT Network Attack Detection using Supervised Machine Learning

Sundar Krishnan *skrishnan@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Ashar Neyaz *ashar.neyaz@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Qingzhong Liu *liu@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Abstract

The use of supervised learning algorithms to detect malicious traffic can be valuable in designing intrusion detection systems and ascertaining security risks. The Internet of things (IoT) refers to the billions of physical, electronic devices around the world that are often connected over the Internet. The growth of IoT systems comes at the risk of network attacks such as denial of service (DoS) and spoofing. In this research, we perform various supervised feature selection methods and employ three classifiers on IoT network data. The classifiers predict with high accuracy if the network traffic against the IoT device was malicious or benign. We compare the feature selection methods to arrive at the best that can be used for network intrusion prediction.

Keywords: Supervised Learning, Network Attack Detection, IoT, Network Forensics, Network Security.

1. INTRODUCTION

Network traffic has seen unprecedented growth in the last decades. With growing volumes of Internet-connected devices, cheaper cloud storage, growing smartphone technology, decreasing device and network hardware costs, and the advent of 5G technology, it is predicted that by 2023, there will be 3X more networked devices on earth than humans. A Cisco Annual Internet Report Forecasts 5G to support more than 10% of Global Mobile Connections by 2023 [1], [2]. This growth in network traffic and Internet-connected devices has resulted in an increase of malicious attacks over the network that can sometimes be difficult to detect. A network attack is a type of cyber-attack in which the attacker attempts to gain unauthorized access into a computer network or an Internet-connected device for malicious purposes or reconnaissance. Cyber-attacks rank as the fastest growing crime in the U.S., causing catastrophic business disruption. Globally, cybercrime damages are expected to reach US $10.5 trillion annually by 2025. NIST defines a cyber-attack (breach) as, “An attack, via cyberspace, targeting an enterprise’s use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information” [3]. Over the years, Cybercrime has moved on from targeting and harming people, computers, networks, and smartphones - to cars, power-grids, smart devices, and anything that can connect back to the Internet.

The Internet of Things (IoT) has come a long way since the 80s when early IoT designers (students) at Carnegie Melon University installed micro-switches inside of a Coca-Cola vending machine to remotely check on the temperature and availability of their favorite beverages [4]. IoT devices and technology have gone mainstream these days, with IoT devices remotely controlling our home speakers, smart elevators, cars, household appliances, power plants, security cameras, baby cams, smart buildings, medical devices, freight, etc. These devices connect back to the Internet via traditional copper wires, fiber, and telecom technology for remote control functionality, thereby making them game for malicious actors using the Internet. IoT devices are often shipped to users with minimal logon security, operating system vulnerabilities, and overall poor security design. This can be mostly attributed to keeping costs down, ease of use for the user, and inadequate security foresight by the manufacturers. Consequently, the attack surface of IoT devices has greatly grown, triggering security and privacy concerns. The infamous Mirai botnet [5] self-replicated by seeking out hundreds of thousands of home routers with weak or non-existent passwords. The roll-out of the 5G mobile networks may further embolden IoT cyber attackers due to the advantage of high bandwidth, ultra-low latency, and fundamentally new networking capabilities of 5G technology [6].

IoT tangibly solves many business problems across industries such as healthcare, smart cities, building management, utilities, transportation, and manufacturing. About 30% of devices on enterprise networks today are network-connected IoT devices [7], making them potential targets over a network. Unlike traditional IT assets like servers and endpoints, these network-connected devices may not be well maintained and documented by IT teams. Such assets can easily be missed from an organization’s proactive security monitoring apparatus. IoT devices are also found in home networks and may not have adequate security controls or infrastructure to protect them. With more and more diverse types of IoT devices continuing to connect to the network, there can be a dramatic broadening of the attack surface. All it takes for a successful intrusion is the diminished integrity of a weak asset on the network.

Predictive capabilities are incredibly beneficial in any industrial setting, especially in thwarting cyber-attacks. Machine learning helps solve tasks (such as regression, clustering, classification, dimensionality reduction, etc.) using an approach/method based on available data. A popular area of machine learning application in cybersecurity is helping businesses detect malicious activity faster and stop attacks before they get started. Cybersecurity should be implemented in layers against any asset. It must be noted that machine learning alone will never be a silver bullet for cybersecurity, but when coupled with other controls, it can improve intrusion detection. While extensive research has been undertaken to predict/detect network attacks on common Information Technology assets, little research has been conducted towards IoT device attacks. In this research, we apply machine learning approaches towards IoT attack detection using the IoTID20 dataset [8] that was built on the network traffic of botnet attacks [9] against IoT devices. Three feature selection models are chosen, and the prediction of an attack based on supervised learning is presented by applying three classifiers against each feature selection model.

2. BACKGROUND

An intrusion detection system (IDS) is a hardware device or software application that monitors a network or host for malicious activity or policy violations [10]. While IDS alerts on intrusions, Intrusion Prevention Systems (IPS) can respond to detected intrusion upon discovery. Intrusion detection using both supervised learning and unsupervised learning has been primarily researched. Using unsupervised machine learning to understand better network attacks has been widely researched. Kumar et al. [11] evaluated MeanShift algorithm to detect network incursion against the KDD99 network traffic dataset. The authors concluded that the MeanShift could detect an attack in the network dataset. However, the algorithm could not detect Remote to Local (R2L) and User to Root (U2R) attacks. Serra et al. Mukherjee et al. [12] proposed ClusterGAN as a new medium for adaptive clustering using Generative Adversarial Networks (GANs). Choi et al. developed a network intrusion detection system (NIDS) using an unsupervised learning algorithm against unlabeled data. The high accuracy of the experiment results provided a recommendation for developing network intrusion detection systems. False attack detection can be challenging to detect. Sakhini et al. [14] evaluated SVM (Support Vector Machine), KNN (K-Nearest-Neighbors), and ANN (Artificial Neural Network) to detect FDI (False Data Injection) attacks. Their experiment results showed that KNN and SVM were more accurate than ANN. Supervised learning is the machine learning task of learning a function that maps an input to an output based on examples (labeled data) of such input-output pairs. Balkanli et al. [17] detected network intrusion with 99% accuracy against 20% of backscatter darknet traffic by employing two opensource network intrusion detection systems (NIDS) and two supervised machine learning techniques on backscatter darknet traffic. Morfino et al. [18] evaluated the performance of various supervised machine learning algorithms in recognizing cyberattacks, specifically, the SYN-DOS attacks on IoT systems by differentiating them in terms of application performances and also in training/application times. Their Apache Spark algorithm yielded an accuracy of greater than 99%, whereas Random Forest achieved an accuracy of 1%. A simple type of attack against IoT devices is Denial-of-Service (DoS). The IoT device receives bursts of surplus network traffic rendering it unusable or overtaxing IoT hardware and underlying infrastructure. Hodo et al. [19] used Artificial Neural Network (ANN) to detect Denial-of-Service (DoS) of Distributed Denial-of-Service (DDoS) attacks with a 99.4% accuracy in attack detection. Loannou et al. [20] put forward the use of Support Vector Machine (SVM) learning model for detecting deviation within the Internet of Things. The proposed SVM model achieved up to 100% accuracy when evaluated against the unknown data taken from the corresponding network topology with proper training. The model also achieved an 81% accuracy when used under an unknown topology. Often IoT devices are wireless and configured to routers with poor security settings [21]. Grimaldi et al. [22] leveraged supervised machine learning techniques in real-time to identify and detect wireless traffic interference, thereby allowing for isolation and extraction of standard-specific traffic. Anthi et al. [23] presented a three-layer intrusion detection system (IDS) that used a supervised machine learning approach to detect a variety of popular network-based attacks on IoT networks. The proposed system's three core functions' performance resulted in an F-measure of 96.2%, 90.0%, and 98.0%, respectively. This demonstrated that the proposed system could automatically distinguish IoT devices on the network and detect attack types against devices on the network. Artificial neural networks and deep learning approaches can also be used to detect network intrusions. Caron et al. [16] proposed a scalable clustering method called DeepCluster for unsupervised learning of convolutional neural networks or convnets against the ImageNet and YFCC100M datasets. Their results obtained were better than other state-of-the-art approaches by a significant margin. There are limitations to using machine learning to identify network attacks. Xiao et al. [15] examined attack models and IoT security solutions based on machine learning techniques. They concluded that supervised and unsupervised learning sometimes fails to detect the attacks due to oversampling, insufficient training data, and bad feature extraction. In this research, we leverage supervised learning to predict normal and malicious/abnormal network traffic using the IoTID20 dataset [8]. Ullah et al. [8] proposed this dataset, namely IoTID20 that was generated from Botnet traffic against IoT devices [9]. Ullah et al. [8] also utilized this dataset to propose a detection classification methodology. In this article, we choose a different approach compared to Ullah et al. [8] when selecting features and classifiers. We then evaluate these various feature selection approaches against classifier accuracy.

3. FEATURE SELECTION IN MACHINE LEARNING

Machine learning is a branch of computational algorithms designed to emulate human intelligence by learning from the surrounding environment [24]. Machine learning (ML) and Artificial Intelligence (AI) have become dominant problem-solving techniques in many areas of research and industry in the last decade. ML and AI are not the same. While Artificial intelligence is about problem-solving, reasoning, and learning in general; Machine learning is specifically about learning—learning from examples, from definitions, from being told, and from behavior [25]. While working with ML, we typically use datasets (like a database table or an Excel spreadsheet) that contain data for the experiment arranged in columns (features). Each feature, or column, represents a measurable piece of data that can be used for analysis. The below discussion is about a few feature engineering (selection) techniques and supervised learning algorithms employed in our research experiments. Often in a dataset, the given set of features in their raw form does not provide enough, or the most optimal, information to train a Machine Learning model. It may be beneficial to remove unnecessary or conflicting features in some instances, which is known as feature selection or feature engineering. Feature selection is a critical and effective approach to ignoring or retaining certain features on a dataset that do not contribute statistically significantly towards the predicted outcome. Thus, only the most significant subset of features are retained in a model while removing these irrelevant, redundant, and noisy features.

3.1 Filter Methods

Filter methods select features from a dataset independently by relying on features' characteristics, which is often the first step before applying machine learning algorithms. Basic and intuitive filter methods help remove Constant features, Quasi- Constant features, and Duplicated features. A dataset can also include correlated features wherein highly correlated features provide redundant information regarding the target. In such cases, removing one of the two highly correlated features can reduce the dimensionality and noise.

3.2 Sequential Forward Processing

Sequential Forward Processing (or forward feature selection) is a wrapper method that iterates through the set of features while evaluating them using a machine learning algorithm. A preset criterion (k features) is selected, which is the maximum number of features to be reached when starting from zero. The initial starting step is to evaluate all features individually and then select the one that results in the best performance [26]. In the second iteration, we test all possible combinations of the selected feature with the remaining features and retain the pair that produces the best algorithmic performance. Subsequent iterations continue by adding one feature at a time in each iteration until the preset criteria is reached.

3.3 Sequential Backward Processing

Sequential Backward Processing (or backward feature selection) is a wrapper method that iterates through the set of features while evaluating them using a machine learning algorithm. A preset criterion (k features) is selected, which is the maximum number of features to be reached when starting from zero. The initial starting step is to consider all the features in the dataset, followed by a performance evaluation of the algorithm [26]. Similar iterations follow by removing one feature (least significant) at a time producing the best performing algorithm using an evaluation metric. Iterations continue removing feature after feature until the preset criteria is reached.

3.4 Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a feature selection method that fits a model (e.g., linear regression or SVM) and removes the weakest feature (or features) until the specified number of features are reached [27]. RFE requires a specified number of features to keep while eliminating dependencies and collinearity that may exist in the model.

4. SUPERVISED LEARNING

Supervised learning in machine learning and artificial intelligence refers to systems and algorithms that determine a predictive model using labeled data points with known outcomes. The model is learned by training through learning algorithms such as linear regression, random forests, or neural networks. As input data is fed into the model, it adjusts its weights through a reinforcement learning process, ensuring that the model has been fitted appropriately [28]. Supervised learning is often used to create machine learning models for Regression and Classification types of problems. A statistical approach known as regression analysis can be implemented to establish a possible relationship between different variables. Regression analysis consists of a set of machine learning methods that allow predicting a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

4.1 Random Forest

Random Forest (RF) is based on decision trees and is one of the many machine learning algorithms used for supervised learning. There are two main ways for combining the outputs of multiple decision trees into a random forest, 1. Bagging (Bootstrap aggregation) used in Random Forests and 2. Boosting (used in Gradient Boosting Machines). Figure: 1 depicts a random forest. Random Forest implementations are available in many machine learning libraries for R and Python, like Caret (R) [30], Scikit-learn (Python sklearn.ensemble.RandomForestRegressor) [31], and H2O (R and Python) [32].

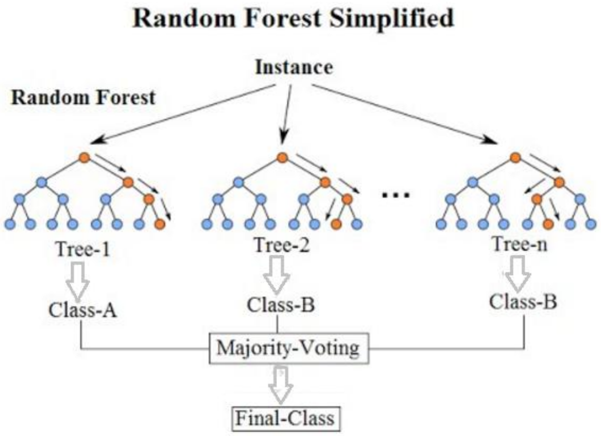

FIGURE 1: A diagram of a random decision forest [29].

4.2 Support Vector Classifier (SVC)

The main task of the algorithm is to find the most correct line, or hyperplane, which divides data into two classes. An SVC is an algorithm that receives input data and returns such a dividing line. In python sklearn library [31], the implementation of SVC is based on libsvm. The objective of a Linear SVC (Support Vector Classifier) is to fit the data and return a “best fit” hyperplane that divides or categorizes the data.

4.3 eXtreme Gradient Boosting (XGBoost)

XGBoost implements machine learning algorithms under the gradient boosting framework and is an optimized end-to-end tree boosting library designed to be highly efficient, flexible, and portable [33]. The XGBoost library implements the gradient boosting decision tree algorithm. Figure: 2 depicts the evolution of XGBoost. Generally, XGBoost is fast when compared to other implementations of gradient boosting [35].

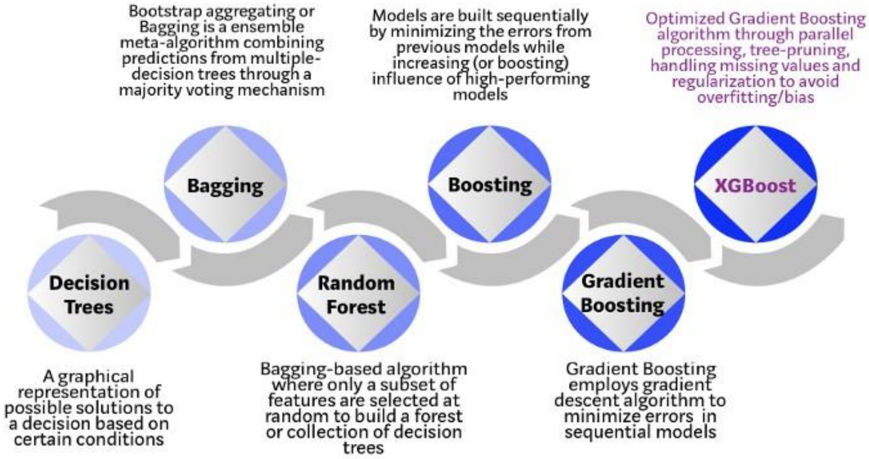
Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

FIGURE 2: Evolution of XGBoost Algorithm from Decision Trees [34].

5. EXPERIMENTS

The IoT dataset in .csv format obtained by Ullah et al. [8] was used for this experiment. The dataset contains network traffic processed from packet captures [9] on two smart home devices wherein attacks on these IoT devices were captured over the wireless network. We decided to focus on the month of May for its ease of use and as it contained all the necessary network attack categories. Python scripts were used for parsing, data preparation, and logistic regression. Results were then documented for analysis. Figure: 3 outlines the workflow of our research.

5.1 Dataset Preparation

Before logistic regression analysis could be performed, few data preparation steps were undertaken below to pre-process and transform the raw data into the necessary data structure to carry out the analysis. The timestamp feature was first formatted for a timestamp format. The dataset was filtered for May/2019 network traffic data using the Timestamp feature. We decided to ignore the features FlowID, Category, and Sub-Category. The feature Label was encoded for Normal =1 and Anomaly=2. The Src IP and Dest IP features were each encoded as 1, 2, 3, and 4 depending on the network class of the IP address values (class A=1, B=2, C=3 and D=4). The Timestamp feature was transformed into Date and Time features (24-hr format). Data rows with invalid Dst IP (0.0.0.x) were ignored. Table 1 shows the features at the end of this step. We used pre-processing techniques such as dropping features that are Constants, Quasi-Constants, and Duplicates. A Pearson's correlation of 0.8 was then applied to further select features. Correlated features degrade the detection capability of a machine learning algorithm, and thus, highly correlated features were ignored from the IoTID20 dataset. Table 2 shows a list of features that were dropped from the IoTID20 dataset at each pre-processing stage and the final set of features to retain at the end of pre-processing.

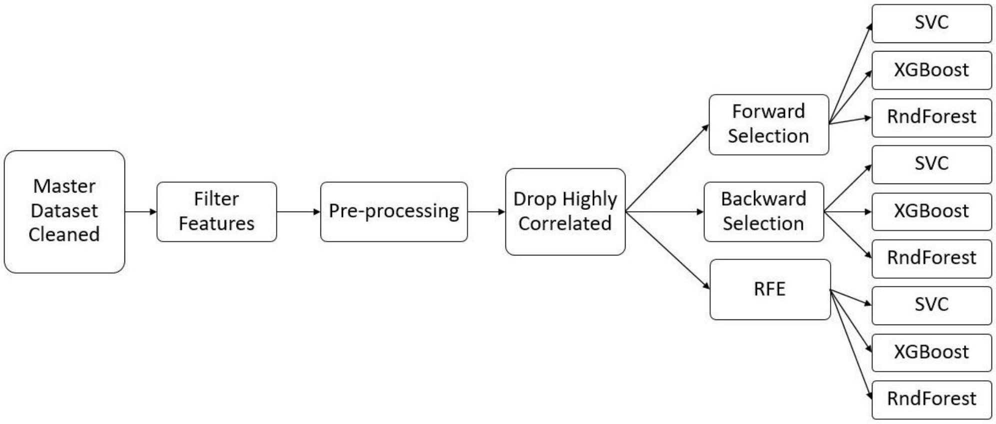

FIGURE 3: Experiment workflow.

5.2 Feature Selection and Logistic Regression

After pre-processing, a separate dataset with 33 features as in Table 3 was created for the experiment. This dataset was then used for the below experiments.

a) Applied Sequential Backward Processing for feature selection and obtained eight features for logistic regression. Table 2 shows the features obtained after applying Sequential Backward Processing. The dataset was split into train/test (80/20) and perform logistic regression using Random Forest Classifier, SVC, and XGBoost. Results were documented for analysis.

b) Applied Sequential Forward Processing for feature selection and obtained eight features for logistic regression. Table 2 shows the features obtained after applying Sequential Forward Processing. The dataset was split into train/test (80/20) and perform logistic regression using Random Forest Classifier, SVC, and XGBoost. Results were documented for analysis.

c) Applied RFE processing for feature selection and obtained eight logistic regression features. Table 2 shows the features obtained after applying RFE. The dataset was split into train/test (80/20) and perform logistic regression using Random Forest Classifier, SVC, and XGBoost. Results were documented for analysis.

Dataset	Features
Original dataset	Src IP,Src IP Cl (network class of Src IP), Src Port, Dst IP, Dst IP CL (network class of Dst IP), Dst Port, Protocol, Timestamp DT (split date value of Timestamp), Timestamp 24HR TIME (split time value of Timestamp), Flow Duration, Tot Fwd Pkts, Tot Bwd Pkts, TotLen Fwd Pkts, TotLen Bwd Pkts, Fwd Pkt Len Max, Fwd Pkt Len Min, Fwd Pkt Len Mean, Fwd Pkt Len Std, Bwd Pkt Len Max, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Flow Byts s, Flow Pks s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Tot, Fwd IAT Mean, Bwd IAT Mean, Fwd IAT Max, Fwd IAT Min, Bwd IAT Tot, Bwd IAT Mean.1, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Len, Bwd Header Len, Fwd Pkts s, Bwd Pks s, Pkt Len Min, Pkt Len Max, Pkt Len Mean, Pkt Len Std, Pkt Len Var, FIN Flag Cnt, SYN Flag Cnt, RST Flag Cnt, PSH Flag Cnt, ACK Flag Cnt, URG Flag Cnt, CWE Flag Count, ECE Flag Cnt, Down Up Ratio, Pkt Size Avg, Fwd Seg Size Avg, Bwd Seg Size Avg, Fwd Byts/b Avg, Fwd Pkts b Avg, Fwd Blk Rate Avg, Bwd Byts b Avg, Bwd Pkts b Avg, Bwd Blk Rate Avg, Subflow Fwd Pkts, Subflow Fwd Byts, Subflow Bwd Pkts, Subflow Bwd Byts, Init Fwd Win Byts, Init Bwd Win Byts, Fwd Act Data Pkts, Fwd Seg Size Min, Active Mean, Active Std, Active Max, Active Min, Idle Mean, Idle Std, Idle Max, Idle Min, Label (converted to binary), Cat, Sub Cat

TABLE 1: Original Dataset after feature cleansing.

Pre-processing / Filter techniques applied	Features dropped as a result
Constant features	Fwd PSH Flags, Fwd URG Flags, Fwd Byts/b Avg, Fwd Pkts b Avg, Fwd Blk Rate Avg, Bwd Byts b Avg, Bwd Pkts b Avg, Bwd Blk Rate Avg, Init Fwd Win Byts, Fwd Seg Size Min, Timestamp month
Quasi-Constant features	Bwd URG Flags, FIN Flag Cnt, RST Flag Cnt, URG Flag Cnt, CWE Flag Count, ECE Flag Cnt
Duplicate Features	PSH Flag Cnt, Fwd Seg Size Avg, Bwd Seg Size Avg, Subflow Fwd Pkts, Subflow Fwd Byts, Subflow Bwd Pkts, Subflow Bwd Byts
Correlated features	Bwd Pkt Len Mean, Idle Mean, Idle Std, Bwd IAT Tot, Fwd Act Data Pkts, Active Min, Fwd Header Len, ACK Flag Cnt, Bwd Pks s, Pkt Len Var, Active Max, Flow IAT Min, Timestamp day, Idle Max, Label, Idle Min, Fwd Pkt Len Mean, TotLen Fwd Pkts, Pkt Len Mean, Flow IAT Max, Bwd Pkt Len Min, Pkt Len Max, Pkt Size Avg, Fwd IAT Min, Bwd IAT Max, Fwd IAT Mean, Timestamp hour, Bwd IAT Min, Fwd IAT Max, Flow Byts s, Bwd IAT Mean.1
Final set of features for experiment	
Src IP Cl, Src Port, Dst IP CL, Dst Port, Protocol, Flow Duration, Tot Fwd Pkts, Tot Bwd Pkts, TotLen Bwd Pkts, Fwd Pkt Len Max, Fwd Pkt Len Min, Fwd Pkt Len Std, Bwd Pkt Len Max, Bwd Pkt Len Std, Flow Pks s, Flow IAT Mean, Flow IAT Std, Fwd IAT Tot, Bwd IAT Mean, Bwd IAT Std, Bwd PSH Flags, Bwd Header Len, Fwd Pkts s, Pkt Len Min, Pkt Len Std, SYN Flag Cnt, Down Up Ratio, Init Bwd Win Byts, Active Mean, Active Std, Label, Timestamp minute, Timestamp second	

TABLE 2: Pre-processing of the IoT20 Dataset.

Feature Selection Applied	Features Obtained
Sequential Backward Processing	’Pkt Len Var’, ’Fwd Header Len’, ’TotLen Fwd Pkts’, ’Timestamp hour’, ’Pkt Len Mean’, ’Fwd Pkt Len Mean’, ’Bwd Pkt Len Min’, ’Timestamp day’, ’Label’
Sequential Forward Processing	’Pkt Len Var’, ’Timestamp hour’, ’Pkt Len Mean’, ’Fwd Pkt Len Mean’, ’Bwd Pkt Len Min’, ’Pkt Size Avg’, ’Timestamp day’, ’Bwd Pkt Len Mean’, ’Label’
RFE	Flow IAT Max’, ’ACK Flag Cnt’, ’Idle Max’, ’Timestamp hour’, ’Fwd Act Data Pkts’, ’Flow IAT Min’, ’Idle Min’, ’Idle Std’, ’Label’

TABLE 3: Feature selection techniques applied for Logistic Regression.

6. ANALYSIS

After pre-processing, the dataset was put to three feature selection processes to arrive at eight highly ranked features. This was followed by three logistic regression algorithms. The first feature selection method was Sequential Backward Processing. The results of SVC, XGBoost, and Random Forest classification against the eight features from Sequential Backward Processing are shown in Table 6. The ROC curve for the three classifiers is shown in Figure: 5, and the Reliability Curve is shown in Figure 4. The second feature selection method was Sequential Forward Processing. The results of SVC, XGBoost, and Random Forest classification against the eight features from Sequential Forward Processing are shown in Table: IV. The ROC curve for the three classifiers is shown in Figure 7, and the Reliability Curve is shown in Figure 6. The third feature selection method was Recursive Feature Elimination (RFE). The results of SVC, XGBoost, and Random Forest classification against the eight features from RFE are shown in Table: IV. The ROC curve for the three classifiers is shown in Figure 9, and the Reliability Curve is shown in Figure 8. From Table 4, we can conclude that all the three supervised feature selection methods could predict with high accuracy malicious traffic vs. benign traffic. The number of features used (eight) was random, but changes to the number used can impact accuracy scores. The Root Mean Square Error (RMSE) is a valuable metric that tells us how far apart our predicted values are from our observed values in a model. The SVC classifier has larger RMSE values in the three feature selection methods, implying a worse model fits the data. Overall, the use of RFE yielded the best accuracy for the three classifiers.

Feature Selection Method	Classifier	Accuracy	F1 Score	Recall	RMSE
Sequential Backward Processing	SVC XGBoost Random Forest	98.20% 99.31% 99.23%	0.98 0.99 0.99	0.97 0.98 0.98	0.134096 0.082918 0.087708
Sequential Forward Processing	SVC XGBoost Random Forest	98.48% 99.30% 99.21%	0.98 1.00 0.99	0.97 0.98 0.98	0.123455 0.083495 0.089068
Recursive Feature Elimination	SVC XGBoost Random Forest	98.76% 99.79% 99.78%	0.98 1.00 1.00	0.98 1.00 1.00	0.111159 0.04599 0.047028

TABLE 4: Logistic Regression Results.

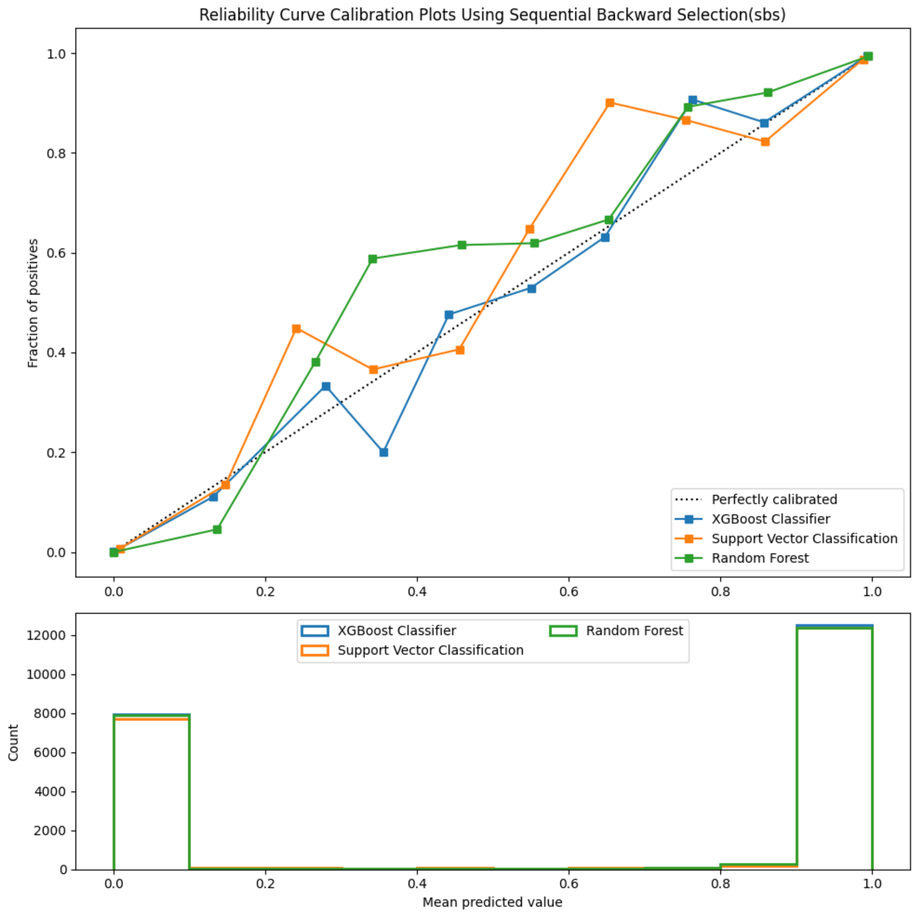

FIGURE 4: Reliability Curve after applying sequential backward feature selection.

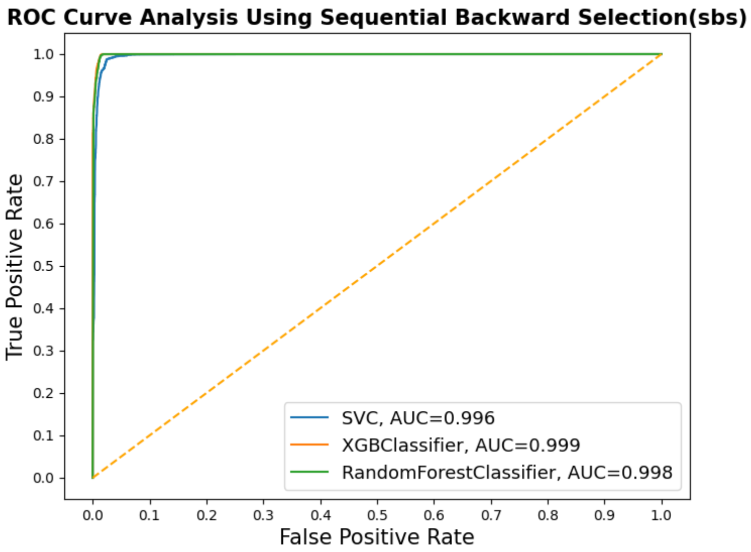

FIGURE 5: ROC after applying sequential backward feature selection.

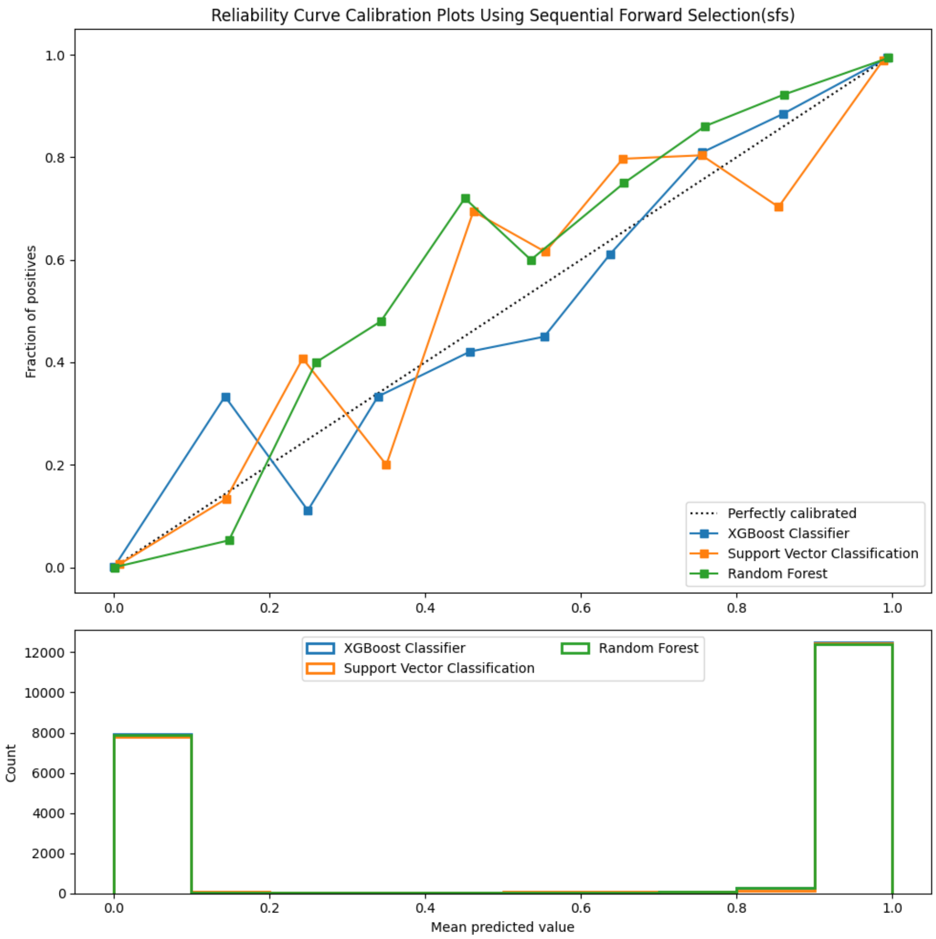

FIGURE 6: Reliability Curve after applying sequential forward feature selection.

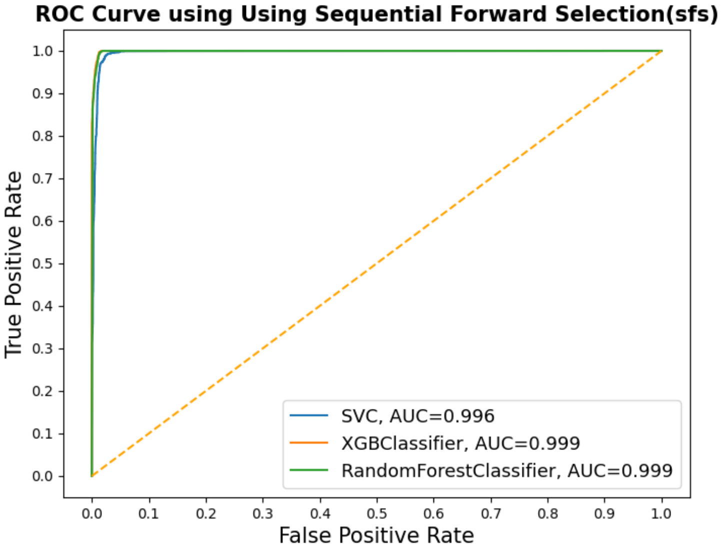

FIGURE 7: ROC after applying sequential forward feature selection.

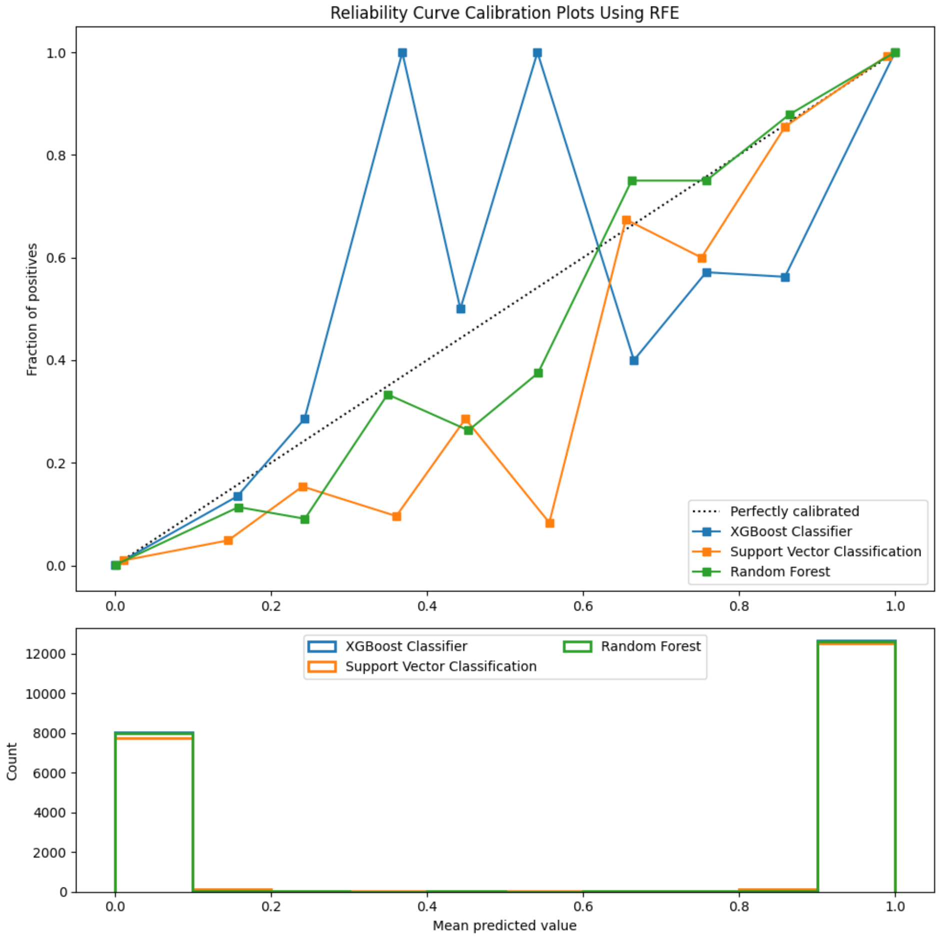

FIGURE 8: Reliability Curve after applying RFE feature selection.

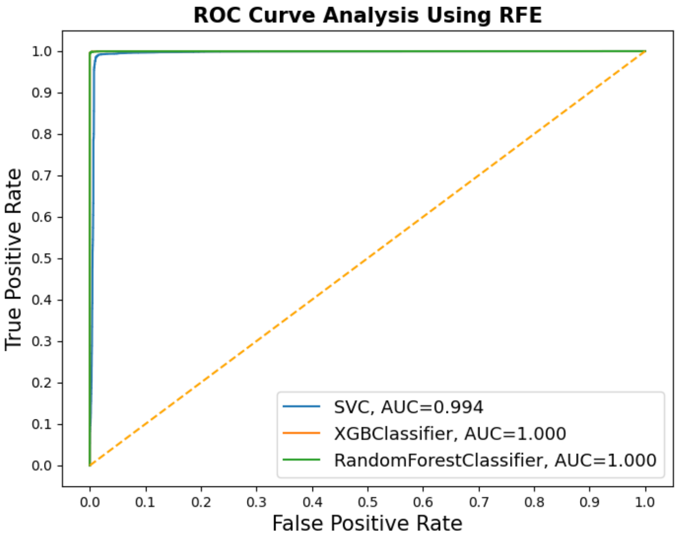

FIGURE 9: ROC after applying RFE feature selection.

7. CONCLUSION

Security threats to IoT systems and devices translate to significant security risks because of the inherent characteristics of the underlying technology. These characteristics make IoT environments versatile, functional, and efficient but can be vulnerable to threat actors. This research evaluates different supervised feature selection methods to predict malicious network traffic against IoT devices. We employ three different feature selection methods and implement three different logistic regression techniques for each selection method. We conclude that all the three logistic regression techniques (SVC, Random Forest, and XGBoost) performed with high accuracy. This implies that these techniques can be employed to predict an attack on IoT devices in a supervised learning setting.

Security attacks on IoT devices can sometimes be challenging to detect since IP addresses can be spoofed by the attacker, making it improper to be used as a machine learning feature. IP addresses are also mostly used in context with other indicators during intrusion detection. While this research can accurately predict an IoT attack, it should be noted that supervised learning is limited to the quality of training data and features selected. Statistical measures for feature selection must be carefully chosen and can significantly impact attack/intrusion predictions. The choice of limiting selection to eight features and limiting the dataset used for the month of May/2019 was purely for the research study. A different choice can result in different classifier accuracy results. Lastly, security against any asset should always be deployed in layers following risk, vulnerability, and threat analysis. A proactive effort by both manufacturers and the business community towards leveraging existing Cybersecurity controls, technology, and industry best practices frameworks can significantly mitigate the fast-rising IoT incident exposures.

8. ACKNOWLEDGEMENT

The authors would like to thank the Cyber Forensics Intelligence Center at Sam Houston State University for allowing the use of its forensic labs.

9. REFERENCES

[1] "New Cisco Annual Internet Report Forecasts 5G to Support More Than 10% of Global Mobile Connect — The Network." Internet: https://newsroom:cisco:com/pressrelease-content?type=webcontentfn&garticleId=2055169, [Jan. 05, 2021].

[2] S. Morgan, "Cybercrime To Cost The World $10.5 Trillion Annually By 2025," Internet: https://cybersecurityventures:com/hackerpocalypse-cybercrime-report-2016/fn#g:f_g:text=Cyberattacksarethefastestgrowing;insizefn%g2Csophisticationandcost:fn&gtext=oeDDoSattacksfn%g2Cransomwarefn%g2Cand;SharkonABC'sSharkTank, 2020, [Jan. 05, 2021].

[3] NIST Glossary, Internet: https://csrc.nist.gov/glossary/term/Cyber_Attack, [Jan. 05, 2021].

[4] "The "Only" Coke Machine on the Internet.", Internet: https://www:cs:cmu:edu/f_gcoke/history long:txt, [Jan. 05, 2021].

[5] Avast Threat Intelligence Team, "Hacker creates seven new variants of the Mirai botnet", Internet: https://blog:avast:com/hacker-creates-seven-new-variants-of-the-mirai-botnet, 2018, [Jan. 05, 2021].

[6] "How 5G and IoT devices open up the attack surface on enterprises - Security Boulevard.", Internet: https://securityboulevard:com/2020/04/how-5g-and-iot-devicesopen-up-the-attack-surface-on-enterprises/, 2020, [Jan. 05, 2021].

[7] P. A. Networks, "2020 Unit 42 IoT Threat Report", Internet: https://start:paloaltonetworks:com/unit-42-iot-threat-report, 2020, [Jan. 05, 2021].

[8] I. Ullah and Q. H. Mahmoud, "A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks," in Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12109 LNAI. Springer, may 2020, pp. 508–520. [On-line]. Available: https://link:springer:com/chapter/10:1007/978-3-030-47358-7 52, [Jan. 05, 2021].

[9] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "IoT network intrusion dataset", Internet: https://ieee-dataport:org/open-access/iot-network-intrusion-dataset, 2019, [Jan. 05, 2021].

[10] "What is an Intrusion Detection System?", Internet: https://www:barracuda:com/glossary/intrusion-detection-system, [Jan. 05, 2021].

[11] A. Kumar, W. Glisson, and H. Cho, "Network Attack Detection Using an Unsupervised Machine Learning Algorithm" in Proc. 53rd Hawaii Int.Conf. Syst. Sci. Hawaii International Conference on System Sciences, 2020, [On-line], Available: https://aisel.aisnet.org/hicss-53/st/cyber_threat_intelligence/8/, [Jan. 05, 2021].

[12] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, Jul 2019, "ClusterGAN: Latent space clustering in generative adversarial networks," in 33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019, [On-line]. vol. 33, no. 01. AAAI Press, pp. 4610–4617.. Available: www:aaai:org, [Jan. 05, 2021].

[13] L. F. Carvalho, S. Barbon, L. D. S. Mendes, and M. L. Proença, Jul 2016, "Unsupervised learning clustering and self-organized agents applied to help network management", Expert Syst. Appl., [On-line]. vol. 54, pp. 29–47, Available: https://dl.acm.org/doi/abs/10.1016/j.eswa.2016.01.032, [Jan. 05, 2021].

[14] J. Sakhnini, H. Karimipour, and A. Dehghantanha, Aug 2019, "Smart Grid Cyber Attacks Detection Using Supervised Learning and Heuristic Feature Selection," in Proc. 2019 7th Int. Conf. Smart Energy Grid Eng. SEGE 2019. Institute of Electrical and Electronics Engineers Inc., [On-line] pp. 108–112., Available: https://ieeexplore.ieee.org/document/8859946, [Jan. 07, 2021].

[15] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, Sep 2018, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?" IEEE Signal Process. Mag., [On-line] vol. 35, no. 5, pp. 41–49, Available: https://ieeexplore.ieee.org/document/8454402, [Jan. 07, 2021].

[16] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, Jul 2018, "Deep Clustering for Unsupervised Learning of Visual Features," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), [On-line] vol. 11218 LNCS, pp. 139–156, Available: http://arxiv:org/abs/1807:05520, [Jan. 07, 2021].

[17] E. Balkanli, J. Alves, and A. N. Zincir-Heywood, Jan 2014, "Supervised learning to detect DDoS attacks," in IEEE SSCI 2014 2014 IEEE Symp. Ser.Comput. Intell. - CICS 2014 2014 IEEE Symp. Comput. Intell. Cyber Secur. Proc. Institute of Electrical and Electronics Engineers Inc., Available: https://ieeexplore.ieee.org/document/7013367, [Jan. 07, 2021].

[18] V. Morfino and S. Rampone, Mar 2020, "Towards Near-Real-Time Intrusion Detection for IoT Devices using Supervised Learning and Apache Spark," Electronics, [On-line] vol. 9, no. 3, p. 444, Available: https://www:mdpi:com/2079-9292/9/3/444, [Jan. 09, 2021].

[19] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, 2016, "Threat analysis of IoT networks Using Artificial Neural Network Intrusion

Detection System” 016 International Symposium on Networks, Computers and Communications (ISNCC), [On-line] pp. 1-6, doi: 10.1109/ISNCC.2016.7746067. Available: https://ieeexplore:ieee:org/abstract/document/7746067/, [Jan. 09, 2021].

[20] C. Ioannou and V. Vassiliou, May 2019, “Classifying security attacks in IoT networks using supervised learning” in Proc. - 15th Annu. Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2019. Institute of Electrical and Electronics Engineers Inc., [On-line] pp. 652–658., Available: https://ieeexplore.ieee.org/abstract/document/8804727, [Jan. 11, 2021].

[21] E. Montalbano, “Report: Most Popular Home Routers Have ‘Critical’ Flaws,” Internet: https://threatpost:com/report-mostpopular-home-routers-have-critical-flaws/157346/, 2020, [Jan. 11, 2021].

[22] S. Grimaldi, A. Mahmood, and M. Gidlund, Dec 2018, “Real-Time Interference Identification via Supervised Learning: Embedding Coexistence Awareness in IoT Devices,” IEEE Access, [On-line] vol. 7, pp. 835–850, Available: https://ieeexplore.ieee.org/document/8570750, [Jan. 11, 2021].

[23] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, Oct 2019, “A Supervised Intrusion Detection System for Smart Home IoT Devices,” IEEE Internet Things J., [On-line] vol. 6, no. 5, pp. 9042–9053, Available: https://ieeexplore.ieee.org/document/8753563, [Jan. 11, 2021].

[24] I. El Naqa and M. J. Murphy, 2015, What Is Machine Learning? Cham: Springer International Publishing, [On-line] pp. 3–11, Available: https://doi:org/10:1007/978-3-319-18305-3 1, [Jan. 14, 2021].

[25] K. Kersting, Nov 2018, “Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines,” Front. Big Data, [On-line] vol. 1, p. 6, Available: https://www:frontiersin:org/article/10:3389/fdata:2018:00006/full, [Jan. 15, 2021].

[26] Y. Charfaoui, “Hands-on with Feature Selection Techniques: Wrapper Methods,” Internet: https://heartbeat:fritz:ai/hands-onwith-feature-selection-techniques-wrapper-methods-5bb6d99b1274, 2020, [Jan. 15, 2021].

[27] “Recursive Feature Elimination — Yellowbrick v1.2.1 documentation.”, Internet: https://www:scikit-yb:org/en/latest/api/modelselection/rfecv:html, [Jan. 15, 2021].

[28] “What is Supervised Learning?”, Internet: https://www:ibm:com/cloud/learn/supervised-learning, [Jan. 16, 2021].

[29] “File:Random forest diagram complete.png - Wikimedia Commons.”, Internet: https://commons:wikimedia:org/wiki/File:Randomforest diagram complete:png, [Jan. 16, 2021].

[30] “Caret: Classification and Regression Training.”, Internet: https://cran:r-project:org/web/packages/caret/index:html, [Jan. 16, 2021].

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, P. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, A. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, and E. Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, “Scikit-learn: Machine Learning in fPgython.” Internet: https://scikit-learn:org/stable/modules/generated/sklearn:svm:SVC:html, [Jan. 18, 2021].

[32] “Distributed Random Forest (DRF).”, Internet: http://docs:h2o:ai/h2o/latest-stable/h2o-docs/data-science/drf:html, [Jan. 18, 2021].

[33] T. Chen and C. Guestrin, Aug 2016, “XGBoost,” Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., Available: http://dx:doi:org/10:1145/2939672:2939785, [Jan. 18, 2021].

[34] V. Morde and V. A. Setty, “XGBoost Algorithm: Long May She Reign!”, Internet: https://towardsdatascience:com/https-medium-com-vishalmordexgboost-algorithm-long-she-may-rein-edd9f99be63d, 2019 [Jan. 18, 2021].

[35] S. Pafka, “Benchmarking Random Forest Implementations”, Internet: http://datascience:la/benchmarking-random-forestimplementations/, 2015, [Jan. 18, 2021].

[36] K. P. Shung, “Accuracy, Precision, Recall or F1?”, Internet: https://towardsdatascience:com/accuracy-precision-recall-orf1-331fb37c5cb9, 2015, [Jan. 18, 2021].