2023 IEEE International Performance, Computing, and Communications Conference (IPCCC) | 979-8-3503-0293-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IPCCC59175.2023.10253835

Anomaly Detection and Attack Classification in IoT Networks Using Machine Learning

Kyungbin Lee, Nahid Ebrahimi Majd
Department of Computer Science and Information System
California State University San Marcos, United States
lee305@csusm.edu, nmajd@csusm.edu

***Abstract*— The rise of network attacks has emerged as a pressing concern for companies and individuals. A Network Intrusion Detection System (NIDS) is employed at the network edge and monitors the traffic exchanged between the network and the cloud. An anomaly detection NIDS detects benign vs attack traffic while a misuse detection NIDS detects benign vs. specific attack. In this paper, we propose two machine learning based NIDS frameworks, an anomaly detection, and a misuse detection NIDS that can effectively detect and classify IoT network attacks. We used Kitsune IoT network attack dataset that contains 9 types of attacks. We used random undersmapling to reduce the dataset size. Then, we employed ANOVA and Chi-square feature selection techniques and studied a variety of machine learning algorithms for binary classification (anomaly detection) and family classification (misuse detection). We studied different models to find the best combination of feature selection method, number of features, and hyperparameters for each model for both binary and family classifications. Our experimental results demonstrated that our proposed Random Forest and Extreme Gradient Boost binary classifiers and Random Forest family classifier with ANOVA feature selection outperform other models and existing research in accuracy.**

***Keywords*— Anomaly Detection; Misuse Detection; IoT; Network Intrusion Detection System; Machine Learning;**

I. INTRODUCTION

Internet of Things (IoT) is widely used on the Internet. International Data Corporation (IDC) estimates that there will be 41.6 billion IoT devices in 2025, capable of generating 79.4 zettabytes (ZB) of data [1]. Some of the most used home IoT devices are security doorbell and baby camera that could be connected to the Internet. These devices record videos that can be accessed via an app on the user's cellphone. The IoT device can send alarms to the app and receive the user's online voice from the cellphone and send it through Internet to the device, which will be played at the device's speaker. In this way, the user can talk to the people who ring their home's doorbell even when the user is far from their home. Such type of IoT devices (e.g., security doorbell) can communicate with other devices on the Internet (in this scenario, the user's cellphone) via bidirectional network traffic streams.

In a factory, the IoT devices can be used to acquire measurements, like level of noise, carbon dioxide emissions, or radioactivity, to prevent harm to the environment, citizens, and industry workers. These devices will collect the acquired measurements to be stored on a server. They alarm the system when an acquired measurement exceeds a certain value. Every piece of equipment in a modern manufacturing plant can report telemetry to the network server [2]. A variety of different types of IoT devices are used by businesses and individuals. Any device that can collect data about its operation is an IoT device, e.g., cellphones, smart watches, and smart cars. They frequently acquire data, which can be analyzed periodically or in time, on the device or on a server.

One IoT device can generate a large volume of data in a short time. As the number of active IoT devices is rapidly growing, the amount of data they collect and transmit/receive to/from the networks is also tremendously growing. However, the average customer usually expects the IoT devices to be low-cost plug-and-play devices with minimum software management and maintenance. Thus, the IoT producers typically employ low-cost processes to produce these devices and employ their software, e.g., unsafe network services and insecure update management process.

As a result, a large percentage of data transmitted on the Internet is generated by IoT devices that are vulnerable to network attacks and security breaches. This results in substantial security concerns where the IoT devices can easily be infected by malwares and unwantedly be the sources of network attacks, especially DDoS attacks that require massive number of always-on Internet-connected devices that can generate extensive amount of data in short time. An infected IoT device might be employed in an active botnet attack even without disturbing any regular operation of the device. That is another reason why the average customer does not pay attention to the security services and updates of their devices while their malware-infected devices can actively help disrupt the Internet.

Due to unique attributes of IoT devices, which are in favor of large scale network attacks, e.g., the large number of devices, heterogeneity of them and the data they produce, always-on nature, ability to generate massive data and freely send it to the network, and poor security protection services, they have become a common target of malwares that originate cyber-attacks, such as botnets and DDoS attacks. In other words, IoT devices have become devices on the Internet that easily get infected by malwares, distribute the infection to other devices, and eventually are employed by attackers as cyber-weapons to run distributed attacks.

Accordingly, the network Intrusion Detection Systems (NIDSs) are located at the network edge and monitor the network traffic arriving or leaving the network. The NIDS detects and blocks any arriving malicious traffic that might infect the devices. An already infected device might be added to network and distribute the infection. NIDS can detect and block malicious traffic circulating in the network as well. Also, the NIDS detects the cyber-attacks originated from IoT devices. Machine Leaning (ML) has recently been used for NIDSs in the networking research community to combat the malicious traffic targeted to or originated from IoT networks [3], [4].

There are two main approaches that an ML based NIDS can implement to fulfill its goals [5].

1. Anomaly Detection (AD)
2. Misuse Detection (MD)

An ML-based anomaly detector is trained only on benign traffic. It can detect benign traffic, and any traffic that is not detected as benign is identified as malicious (anomaly or outlier) traffic [6-8]. A ML-based misuse detector is trained on both benign and malicious traffic. A misuse detector can be either a binary or family classifier. A binary MD classifies benign vs. malicious traffic while a family MD classifies benign vs. specific attack [9-12]. Other approaches have also been proposed: (1) Attack Classification (AC), which assumes the benign traffic is already detected and only classifies different types of attacks; (2) Hybrid, which initially detects malicious traffic vs. benign, and then classifies the type of attack using anomaly detection. An IoT-aware ML-based NIDS uses an IoT dataset extracted from an IoT network traffic [13-18].

The main contribution of this paper is we propose two NIDS frameworks, (1) an anomaly detector binary classifier and (2) a misuse detector family classifier. We used random undersmapling to reduce the dataset size. We studied a variety of ML algorithms with feature selection techniques to reduce the noise in each of our classifiers and get the most accurate model. We also studied the impact of data imbalance in our family classifiers.

The rest of this paper is organized as the following. Section 2 describes our methodology, including our randomly undersmapled dataset and data pre-processing. Section 3 explains our hypertunning. Section 4 describes the performance measures. Section 5 presents the results and discussion. Section 6 draws the conclusion.

II. Methodology

A. Dataset

We used a dataset extracted from Kitsune Network Attack dataset [18], [19]. This dataset has been generated using two physical testbeds, one using IoT cameras and the other with multiple IoT devices in a wireless network. With this topology consisting of two IoT networks, they could perform attacks like SYN flooding attack and Man-In-The-Middle attacks from the second IoT network targeting the cameras in the first network. They infected the IoT devices by different types of network attacks and captured the IoT traffic using Wireshark. Then, they extracted 115 features from the captured traffic. The dataset features are based on aggregated statistics of traffic streams for five timeframes of 100ms, 500ms, 1.5sec, 10sec, and 1min.

The Kitsune dataset is a collection of 9 network attack datasets, containing 2 Reconnaissance attacks, namely (1) OS Scan, (2) Fuzzing, 3 Man-In-The-Middle attacks, namely (3) Video Injection, (4) ARP MitM, (5) Active Wiretap, 3 DoS attacks, namely (6) SSDP Flood, (7) SYN DoS, (8) SSL Renegotiation, and 1 Botnet Attack, namely (9) Mirai. Each of these 9 datasets contains records extracted from benign traffic and one type of attack traffic. In total, the entire dataset consists of more than 27 million records. We ran random sampling to extract a subset from each of the 9 original datasets.

Binary classification: To prepare a dataset for binary classification models, from each of our 9 sub-datasets, we sampled 6,500 benign and 6,500 malicious records. Then, we split each sub-dataset into train and test datasets with the ratio of 70:30. Then, we merged the training sub-datasets and test sub-datasets to get the train set and test set. In total, we had a dataset of 117,000 records, 58,500 benign and 58,500 malicious samples, split to train set (81,900 records) and test set (35,100 records). We used this dataset for our binary classification models.

Family classification: To prepare a dataset for family classification models, from each of our 9 sub-datasets, we sampled 795 benign and 6,500 malicious records. In total, we had 7,150 benign records and 6,500 records of each attack type. Then, we split each sub-dataset into train and test datasets with the ratio of 70:30. Then, we merged the training sub-datasets and test sub-datasets to get the train set and test set. In total, we had 9,100 records in the train set, and 3,900 records in the test set. We used this dataset for our family classification models.

B. Data preprocessing

There were no missing values or duplicates in the dataset, therefore, minimal preprocessing was required. All attributes were numerical, therefore, no encoding was required. We did not create a validation dataset since we used 5-fold cross validation to generalize the models. We used normalization technique to convert each of the variables into a similar scale by centering each variable at zero with a standard deviation of 1.

After the data preprocessing step, we applied feature selection techniques to train the dataset and fed the transformed data to various ML algorithms. For binary classification we labeled the records either 0 for benign or 1 for attack (any type). For family classification, we labeled the records either 0 for benign or a number in range [1-9] where each number identifies one type of attack in order. Fig. 1 illustrates the 2 classes of our binary classifications. Fig. 2 illustrates the 10 classes of our family classifications.

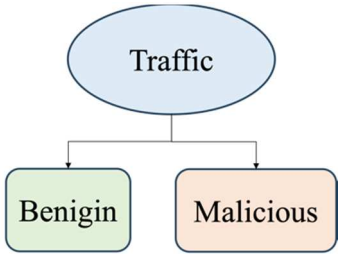

Fig. 1. Two classes of binary classification

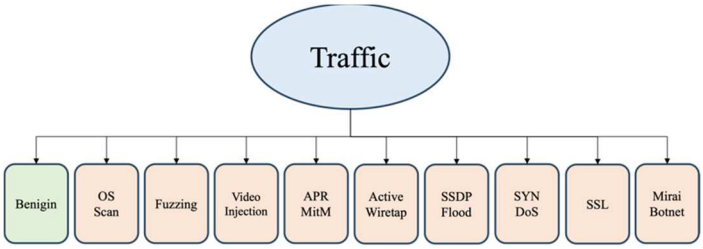

Fig. 2. Ten classes of family classification

Table 1: The top 6 features with highest F-scores

No	Feature number	Feature name	F-score
1	Feature 49	HH_L1_radius	13,502
2	Feature 42	HH_L3_radius	13,496
3	Feature 35	HH_L5_radius	13,408
4	Feature 40	HH_L3_std	12,821
5	Feature 47	HH_L1_std	12,805
6	Feature 33	HH_L5_std	12,764

Table 2. The dataset notation for each aggregation basis

Basis of Aggregation	Acronym	Description
Source IP	H	Traffic from an IP
Source MAC-IP	MI	Traffic from an IP and a specific gateway
Channel	HH	Traffic of a channel between 2 hosts
Socket	HpHp	Traffic between 2 hosts at specific port numbers
Network jitter	HH_jit	A subtype of Channel considering the time interval between 2 packets

III. The Feature Selection Techniques

In this research, we studied two feature selection techniques: ANOVA (Analysis of Variance), and Chi-square. We studied different numbers of features for each of these two feature selection methods on different machine learning models. ANOVA computes an F-score for each feature. A higher F-score represents a higher correlation of the feature with the label. Table 1 presents the list of top 6 features with highest F-scores and their F-score values.

In this dataset, to extract the features, the traffic streams have been aggregated based on the items given in Table 2. For instance, H_L0.1_mean feature represents the statistical mean obtained from the last 10 seconds timeframe of the traffic stream from a specific source IP. For each traffic stream, they aggregated the most recent packets in 5 different timeframes and computed 23 aggregated statistics of the stream for each timeframe. The 5 timeframes are 100ms, 500ms, 1.5sec, 10sec, and 1min, which are denoted by L5, L3, L1, L0.1 and L0.01, respectively. In total they extracted 23x5=115 features.

In this dataset, "HH" represents a stream of packets from a source IP to a destination IP, "HH_radius" represents the root squared sum of the variances of the packets in a stream, and "HH_std" represents the standard deviation sum of the packets in a stream. The "L" represents the time frame length where the recent packets of the stream have been captured. The number after "L" is the decay factor Lambda used in the damped window. For example, L5, L3, and L1 indicate short time frames of 100ms 500ms, and 1.5sec vs L0.01, which indicates a long time frame of 1min.

Table 3: The tunned hyperparameters for each model

No	Model	Tunned Hyperparameters
1	LG	C=100, penalty='none', solver='newton-cg'
2	DT	criterion='entropy', max_depth=20, min_sample_leaf=9
3	RF	criterion='entropy', max_depth=20, n_estimators=90
4	GB	learning_rate=0.1, n_estimators=90
5	SVM	C=1000, kernel='rbf', gamma=3.0
6	KNN	n_neighbors=6
7	XGB	learning_rate=0.1, n_estimators=90
8	NB	priors='none', var_smoothing=0.1
9	ET	criterion='entropy',max_depth=90, n_estimators=30

The top F-scores in Table 1 indicate that the features that have the highest impact on the label are the ones that represent the radius and standard deviation of streams in the most recent time frame of 100ms, 500ms or 1.5 sec. It appears the longer time frames of 10sec or 1min or other stream statistics, like the number of packets or the mean size of them in a stream have less impact on the label identification.

It worth noting that although we present the top 6 F-scores here, we did not observe a big gap between the F-scores of all features, meaning almost all features have some impact on the label identification. However, our results showed that there is a certain number of features that results in minimum noise and highest accuracy in each model. For some models, reducing the number of features reduced the accuracy. Thus, after studying different numbers of features, we used all features to build those models to get the most accurate model.

IV. Hyperparameter Tunning

We performed ANOVA and Chi-square feature selection methods and then applied 9 machine learning algorithms containing Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), k-Nearest Neighbor (KNN), Extreme Gradient Boost (XGB), Gradient Boosting (GB) and Extra Trees (ET) for binary classification and 5 algorithms containing DT, RF, SVM, KNN, NB for family classification.

We tunned hyperparameters to find the best combination of feature selection method, number of features, and hyperparameters for each model for both binary and family classifications. We used grid search for hypertunning. Table 3 illustrates a list of tunned hyperparameters that we optimized for our proposed binary classifier models.

For family classification also, we tried to tune the hyperparameters, but it appeared that the same values as given in Table 3 present the best results.

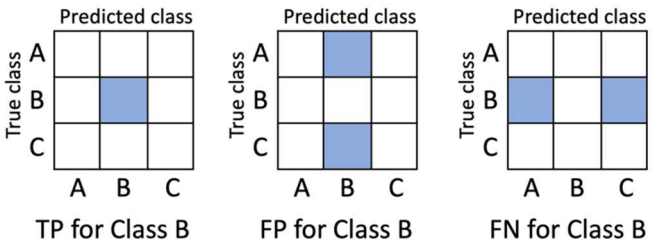

Fig. 3. Confusion matrix in family classification

V. Performance Measures

To analyze the effectiveness of machine learning algorithms, it is important to understand the metrics used to measure the performance of these models.

1. TP (True Positive): An instance of class B was correctly predicted to be in class B.
2. FP (False Positive): An instance of another class was incorrectly predicted to be in class B.
3. FN (False Negative): An instance of class B was incorrectly predicted to be in another class.

Accuracy: Accuracy measures the proportion of the correctly predicted instances of a class to the total number of predictions in that class. It is measured by equation 1.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

Precision: Precision measures the proportion of the correctly predicted instances of a class to the total number of instances that were predicted to be in that class, either correctly or incorrectly. It is measured by equation 2.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall: Recall measures the proportion of the correctly predicted instances of a class to the total number of instances in that class that were provided. It is measured by equation 3.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1 score: F1 score measures the 'Harmonic mean' of precision and recall values. It is measured by equation 4.

$$F1\ score = \frac{2\times(precision\times recall)}{precision + recall} \quad (4)$$

VI. Results and Discussion

For both binary and family classifications, we applied both ANOVA and Chi-square feature selection methods with a variety of feature numbers, and for each one, we tunned hyper parameters. Table 4 illustrates our best binary classification models and Table 5 presents our best family classification models for each algorithm.

In Tables 4 and 5, "None" for feature selection column indicates that we have tried the two features selection methods with different feature numbers but feature selection has reduced the accuracy, thus we used all features to get the most accurate classifier.

Among the binary classification models of Table 4, RF and XGB present the same highest accuracy of 96.9%. RF presents less False Positives but more False Negatives. ET with accuracy of 96.4% and DT with accuracy of 96% present comparable performances. To better investigate the results, we plotted the ROC chart for these models. Fig. 3 illustrates the ROC chart for our best binary classifiers. The chart indicates that RF and XGB with AUC of 0.992 present the best performance among these models.

Among the family classification models of Table 5, RF presents the highest accuracy of 98%. DT and SVM with accuracies of 97.1% and 96.5% respectively are the next most accurate models. To better investigate the results, we plotted the ROC chart for these models. Fig. 4 illustrates the ROC chart for our best family classifiers. The chart indicates that RF, DT, and SVM with AUC of near 1.0 present the best performances among these models. We select RF as our best family classification model with highest accuracy.

We investigated our family classification RF model with more details. Fig. 5 depicts the confusion matrix of this model. Table 6 illustrates the results of this model per class. All 9 attacks demonstrate high True Positive predictions. A few benign instances are Falsely predicted to be attack (row 0, columns 1-9), which causes low recall for benign traffic. A few attack instances (especially 102 Mirai instances) are Falsely predicted to be benign (column 0, rows 1-9), which caused low precision for benign class and low recall for Mirai class. We observed the same pattern in our other classifiers as well.

This could be due to similarities between Mirai and Benign traffic patterns. To test it, we doubled the number of samples for Mirai and Benign classes. This change increased Mirai's recall but decreased Mira's precision, and overall, the model presented lower accuracy. We trained models with higher numbers of benign records, but that reduced the model's accuracy in classifying attacks, and the trained model misclassified the attacks to other types of attacks. We concluded the most accurate model is achieved using a balanced dataset.

[18] proposed Kitsune, an online anomaly detection framework using an ensemble of autoencoder neural networks. They stacked their network, so that the output reconstruction error (RE) of each member is an input the output autoencoder. They also created Kitsune dataset [19] to evaluate their Kitsune model. They reported ROC-AUC ranging from 0.58 to 0.99 for their model. [20] proposed a framework using feature selection techniques to detect IoT attacks. They used Kitsune dataset [19] to evaluate their framework. They sampled 6,500 instances of each class from the original dataset to form their dataset. However, they excluded Mirai and Video Injection attacks from their dataset. They studied a variety of feature selection methods and ML algorithms, including Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (KNN), and Stacking Ensemble of DT, RF, KNN, and meta regressors using logistic regression. We compare our results with this research. We used the same Kitsune dataset with the same number of undersampled records. Unlike [20] that excluded Mirai and Video Injection attacks, we included all 9 types of attacks. They studied only family classification, but we studied both binary and family classifications.

Table 4: The results of the proposed binary classification models.

No	Model	Feature selection, No of Features	Accuracy	Precision	Recall	F1-Score	FP	FN	TP	TN
1	GB	chi-squared, 100	0.938	0.96	0.92	0.94	697	1,479	16,094	16,830
2	NB	ANOVA, 60	0.640	0.97	0.36	0.52	193	11,310	6,263	17,334
3	LG	None, all features	0.796	0.89	0.68	0.77	1,514	5,640	11,933	16,013
4	KNN	ANOVA, 80	0.909	0.91	0.91	0.91	1532	1662	15,911	15,995
5	SVM	None, all features	0.948	0.94	0.95	0.95	1093	847	16,726	16,434
6	DT	ANOVA, 90	0.960	0.95	0.97	0.96	832	560	17,013	16,695
7	ET	ANOVA, 100	0.964	0.96	0.97	0.96	666	600	16,973	16,861
8	XGB	ANOVA, 100	**0.969**	0.96	0.98	0.97	755	342	17,231	16,772
9	RF	ANOVA, 90	**0.969**	0.96	0.98	0.97	722	389	17,184	16,805

Table 5: The results of the proposed family classification models comparing to [20].

No	Model	Feature selection, No of Features	Accuracy	Precision	Recall	F1-Score	AUC
1	NB	None, all features	0.473	0.54	0.48	0.42	0.749
2	KNN	ANOVA, 80	0.926	0.93	0.93	0.93	0.994
3	SVM	None, all features	0.965	0.97	0.97	0.97	1.0
4	DT	ANOVA, 90	0.971	0.97	0.97	0.97	0.999
5	RF	ANOVA, 90	**0.980**	0.98	0.98	0.98	1.0
6	SE [20]	None, all features	0.9726	0.9761	0.9699	0.973	0.9765

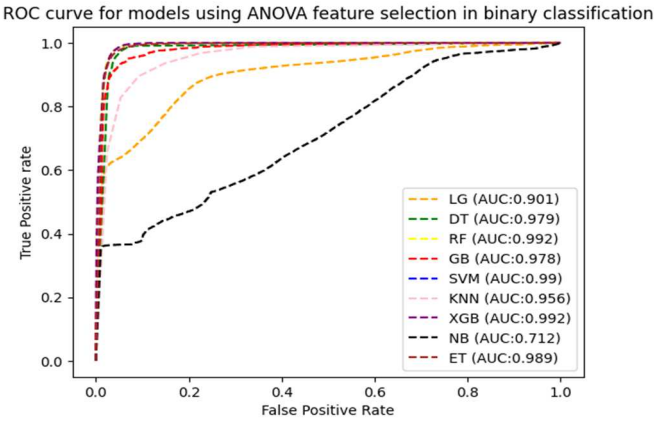

Fig. 4. ROC chart for binary classification models

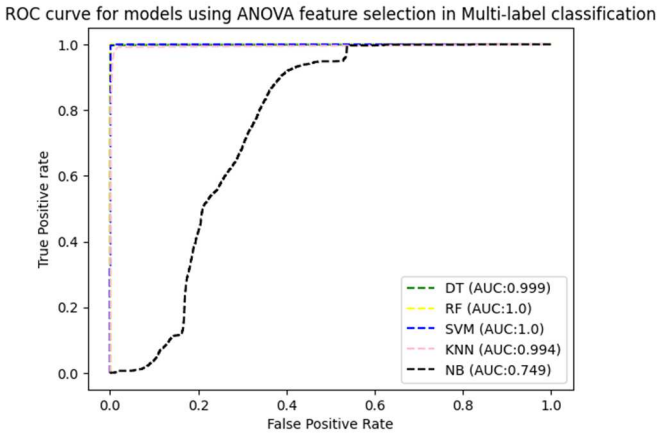

Fig. 5. ROC chart for family classification models

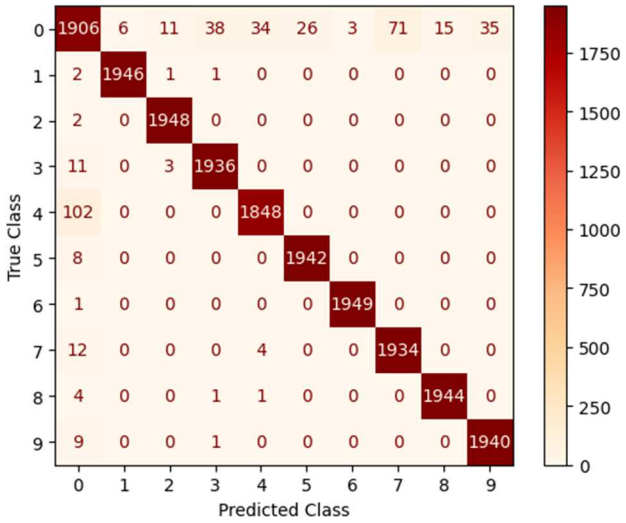

Fig. 6. Confusion Matrix for RF model

Table 6: RF results per each class

Class No	Class Name	Precision	Recall	F1-Score
0	Benign	0.93	0.89	0.91
1	Active Wiretap	1.00	1.00	1.00
2	ARP MitM	0.99	1.00	1.00
3	Fuzzing	0.98	0.99	0.99
4	Mirai	0.98	0.95	0.96
5	OS Scan	0.99	1.00	0.99
6	SSDP Flood	1.00	1.00	1.00
7	SSL Renegotiation	0.96	0.99	0.98
8	SYN DoS	0.99	1.00	0.99
9	Video Injection	0.98	0.99	0.99

The results of [20] showed that their proposed Stacking Ensemble family classifier with no feature selection method (when all features are used to train the model) presents the best performance with 97.26% accuracy, 97.61% precision, 96.99% recall, and 97.65 AUC. Our results showed that our proposed RF family classifier model with ANOVA feature selection where 90 features are selected presents the best performance with 98% accuracy, precision, and recall and 100% AUC. Comparing to [20], our model presented higher accuracy, precision, recall, and AUC, with ability to classify all 9 classes.

VII. Conclusion

With the increasing threat of various types of network attacks, it is essential to develop a system that can effectively detect known and new forms of network attacks. In this research, we designed machine learning models and performed extensive experiments by filtering feature selection methods along with hyperparameter tuning to achieve the best results. The experimental results demonstrates that among binary classification models, our proposed RF and XGB with 96.9% accuracy perform the best, and among family classification models, our proposed RF model with 98% accuracy is the most accurate model. We studied different sizes of samples for misclassifies classes, however, our results indicated that a balanced dataset presents the best performance for this dataset.

In conclusion, our research demonstrated the importance of selecting the appropriate machine learning algorithm and feature selection, and tunning the hyperparameters. Also, our studies showed the importance of balanced number of samples in the dataset in keeping the accuracy high.

References

[1] Available online, https://infohub.delltechnologies.com/l/edge-to-core-and-the-internet-of-things-2/internet-of-things-and-data-placement

[2] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," IEEE Access, 2020. doi: 10.1109/ACCESS.2020.2992249.

[3] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," IEEE access, 2018. doi: 10.1109/ACCESS.2018.2841987.

[4] A. S. Dina and D. Manivannan, "Intrusion detection based on machine learning techniques in computer networks," Internet of Things, 2021. doi: 10.1016/j.iot.2021.100462

[5] C. A. de Souza, C. B. Westphall, R. B. Machado, L. Loffi, C. M. Westphall, and G. A. Geronimo, "Intrusion detection and prevention in fog based IoT environments: A systematic literature review," Computer Networks, 2022. doi: 10.1016/j.comnet.2022.109154.

[6] Y. Xu, L. Zhang, B. Du, L. Zhang, "Hyperspectral anomaly detection based on machine learning: An overview," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022. doi: 10.1109/JSTARS.2022.3167830.

[7] H.W. Oleiwi, D.N. Mhawi, H. Al-Raweshidy, "MLTs-ADCNs: Machine learning techniques for anomaly detection in communication networks," IEEE Access, 2022. doi: 10.1109/ACCESS.2022.3201869.

[8] D. Kim, T.Y. Heo, "Anomaly detection with feature extraction based on machine learning using hydraulic system IoT sensor data," Sensors, 2022. doi: 10.3390/s22072479.

[9] Q. Schueller, K. Basu, M. Younas, M. Patel, and F. Ball, "A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center," IEEE ITNAC, 2018. doi: 10.1109/ATNAC.2018.8615255.

[10] S.S. Sugi, S.R. Ratna, "Investigation of machine learning techniques in intrusion detection system for IoT network," 3rd international conference on intelligent sustainable systems (IEEE ICISS), 2020. doi: 10.1109/ICISS49785.2020.9315900.

[11] Y.K. Saheed, A.I. Abiodun, S. Misra, M.K. Holone, R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," Alexandria Engineering Journal, 2022. doi: 10.1016/j.aej.2022.02.063.

[12] Y. Rbah, M. Mahfoudi, Y. Balboul, M. Fattah, S. Mazer, M. Elbekkali, B. Bernoussi, "Machine learning and deep learning methods for intrusion detection systems in iomt: A survey," 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IEEE IRASET) 2022. doi: 10.1109/IRASET52964.2022.9738218.

[13] R. Zhao, Y. Wang Y, Z. Xue, T. Ohtsuki, B. Adebisi, G. Gui, "Semi-supervised federated learning based intrusion detection method for internet of things," IEEE Internet of Things Journal, 2022. doi: 10.1109/JIOT.2022.3175918.

[14] J. Li, Z. Zhao, R. Li, H. Zhang, and T. Zhang, "AI-based two-stage intrusion detection for software defined IoT networks," IEEE Internet Things Journal, 2018. doi: 10.1109/JIOT.2018.2883344.

[15] I. Ullah and Q. H. Mahmoud, "A two-level hybrid model for anomalous activity detection in IoT networks," IEEE CCNC, 2019. doi: 10.1109/CCNC.2019.8651782

[16] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, A. Pescapé, "A hierarchical hybrid intrusion detection approach in IoT scenarios", IEEE GLOBECOM, 2020. doi: 10.1109/GLOBECOM42002.2020.9348167

[17] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baIoT —network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, 2018. doi: 10.1109/MPRV.2018.03367731.

[18] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," Network and Distributed Systems Security Symposium (NDSS), 2018.

[19] Kitsune Network Attack Dataset. 2019. UCI Machine Learning Repository. doi: 10.24432/C5D90Q.

[20] Y.E. Kim, Y.S. Kim, and H. Kim H, "Effective feature selection methods to detect IoT DDoS attack in 5G core network. Sensors," 2022. doi: 10.3390/s22103819.