Journal of Scientific & Industrial Research
Vol. 83, August 2024, pp. 864-872
DOI: 10.56042/jsir.v83i8.7437

Performance Evaluation of Parametric and Non-Parametric Machine Learning Models using Statistical Analysis for RT-IoT2022 Dataset

Sharmila B S[1]*, Nandini B M[2], Kavitha S S[1] & Anand Srivatsa[1]

[1]Department of Electronics and Communication Engineering, [2]Department of Information Science and Engineering, The National Institute of Engineering, Mysuru 570 008, Karnataka, India

Received 19 December 2023; revised 18 April 2024; accepted 14 June 2024

With the vast growth of the Internet of Things (IoT) applications, the number of devices connected to the IoT is increasing exponentially. On the other hand, cybercriminals are generating sophisticated new cyber attacks to exploit IoT devices. However, conventional Intrusion Detection Systems (IDS) that rely on an alert-based approach fail to detect these novel attacks. Machine Learning (ML) based IDS has the potential to spot even small mutations and new threats. This study investigates the statistical tests like Kolmogorov-Smirnov (KS) test, skewness test, kurtosis test, Pearson's Correlation Coefficient (PCC) and Information Gain Ratio techniques on the recently introduced RT-IoT2022 dataset. The aim is to determine the optimal machine learning algorithms for detecting vulnerabilities within this dataset. The results of skewness and kurtosis tests identify the features having outliers and the KS test indicates that the proposed dataset exhibits non-parametric characteristics. Subsequently, Pearson's Correlation Coefficient (PCC) and Information Gain Ratio techniques are applied to analyze the correlation between features and the categories of the target attacks. Further, parametric and non-parametric ML models are tested to validate the results of statistical tests. With the non-parametric Decision Tree algorithm achieving the highest accuracy of 99.85% among all other models, we conclude that non-parametric ML models are optimal for detecting mutant vulnerabilities.

Keywords: Dataset, Feature extraction, IDS, Internet of things, Machine learning

Introduction

The Internet of Things (IoT) is an emerging technological paradigm. The IoT has the potential to improve many aspects of our daily life.[1,2] Industrial robots, autonomous vehicles, intelligent houses, an even smart cities are all examples of this technology.[3] Unfortunately, When a technology receives widespread acclaim and deployment, it becomes a target for hackers may attempt to abuse and exploit utilizing any number of sophisticated hacking tactics.[4]

Attackers employ a range of hacking strategies to target weak and unencrypted IoT devices to accomplish their objectives, such as cyber espionage and corrupting IoT resources, which pose a potential threat to the sustainability of the IoT.[5] However, the traditional security solutions deployed on IoT devices are more dangerous to cyberattacks because of their limited CPU speed, memory, and energy consumption.[6] The Forensic experts utilises the cyber tools include Intrusion Detection Systems (IDS). IDS is a monitoring tool that continuously scans the network to identify harmful activity. Traditional Intrusion Detection System (TIDS) rely on known intrusion signatures to identify attacks on a network. However, conventional IDS creates massive false alerts against network traffic behavior for even a minor divergence.

A significant number of people in the cybersecurity industry have been paying attention to the progress of Artificial Intelligence.[7] Therefore, advanced security measures need to be implemented for cyber forensics to reduce the consequences. In the cyber field, the ability of AI to detect tiny mutations and novel attacks helps to implement advanced IDS.[8,9] However, a realistic and well-organized dataset is necessary for training and validating the accuracy of an AI-based IDS model. In this research work, we discuss and analyse the novel RT-IoT2022 dataset. The dataset's novelty comes from diverse network traffic from real-time IoT devices like Amazon Alexa, Think-Speak-LED, MQTT-Temp, and Wipro Bulbs. This dataset also includes significant IoT attack types such as SSH Brute Force, DDoS, Nmap, and ARP Poisoning attacks.

On the other hand, advanced machine-learning libraries make it easy to apply a wide range of

*Author for Correspondence
E-mail: sharmilabs@nie.ac.in

machine-learning models to any predictive modeling dataset. Choosing the appropriate model from a large group is thus a real challenge in the field of AI.[10] To select suitable predictive models, first, we need to study the nature of the dataset. An initial step in data analysis is to test for normality. The term "normality" refers to a continuous probability distribution having symmetry about the mean.[11] If data follows a normal distribution, parametric machine learning algorithms are applied; otherwise, nonparametric machine learning models needs to be selected. Different statistical tests are available to test the dataset. In this research work, we utilized the Kolmogorov-Smirnov test[12] to analyze the dataset for normality and also compare the distribution of training and testing datasets. Further, the dataset is subjected to skewness and Kurtosis test to check the asymmetry and to identify the redundancy.[13]

The larger dataset under investigation in this study increases the computational complexity of ML models. In addition, we need to use supervised and unsupervised ML models to detect known and unknown threats. Hence, in this study, we employed Pearson's Correlation Coefficient (PCC) for unsupervised models and the Information Gain Ratio (IGR) for supervised models to extract vital features.[14] Finally, the objective is to identify the most effective ML algorithms for vulnerability detection within this dataset.

Literature Survey

This section investigates publicly accessible state-of-art datasets to demonstrate the limitations and emphasize the need for extensive and intelligent IoT network traffic datasets.[15] The MIT Lincoln Laboratory developed the benchmark KDDCUP99 dataset for analyzing forensics and network security. It is an upgraded version of the DARPA (Defense Advanced Research Project Agency) dataset containing 4.9 GB of records. The dataset includes benign and attacks variants, such as DoS, U2R, R2L, and Probe. The lack of recent malware attacks and the significant amount of redundancy in the dataset prevents its use in the present technological context.

As part of the Australian Centre for Cyber Security project, Nour *et al.,* proposed the UNSW-NB15 dataset developed at Cyber Range Lab.[16] This dataset has three primary servers, two for disseminating legitimate traffic and one for producing attack traffic. The IXIA Perfect Storm tool simulated network traffic for 31 hours, gathering 2,540,044 traces, including nine attack families and legitimate traffic. The Bro-IDS tool extracts the flow-based characteristics of the dataset containing 41 features. After a comprehensive study, it was discovered that the UNSB-NB15 dataset has a significant imbalance, with 87% of the records being of the Normal category and only 0.007% being of the Worms sub category. Furthermore, most attack classes share characteristics with legitimate traffic, leading to class overlap in the dataset.[17]

Sharafaldin *et al.,*[18] suggested the CICIDS2017 dataset comprises both benign and malicious traffic released in 2017. The dataset includes six attack profiles containing DDoS, Web, Infiltration, DoS, Brute Force, and DoS attacks. The target network is equipped with firewalls, routers, switches, and operating systems that resemble home computers. However, the CIC flow meter's ability to filter IPv4 traffic and ignore IPv6 data limits its applicability as a bidirectional feature extractor for IPv6-based IoT devices.

Bot-IoT is a realistic dataset created in UNSW Canberra's Research Cyber Range lab for IoT network security scenarios.[19] The framework includes Kali computers, Windows 7, and Ubuntu PCs. With the help of the Ostinato tool, the normal traffic is simulated, and JavaScript for Node-red middleware mimics IoT sensors. The Argus tool extracts features from both normal and malicious traffic. Since traffic generated by IoT devices had established itself in the simulated environments, testing IDS systems and network forensics with this data doesn't accurately reflect real-world validation of IDS.

The HIKARI-2021 dataset attempts to record HTTPS protocol traces of application layer attacks. This dataset represents common application layer attacks like probing and brute force.[20] As a part of preprocessing stage, the Zeek tool transforms packet-based traffic into bi-directional flow-based traffic. Since the normal and malicious datasets are synthetic, they are inadequate to represent actual network activity.

The latest Aposemat IoT-23 dataset, proposed by Parmisano *et al.,*[21] represent the IoT network traffic. In Aposemat IoT-23, real IoT devices such as the Philips HUE lamp, Amazon Echo, and Somfy Smart Door Lock generate IoT network traces communication. The stratosphere project analyzed twenty different variants of malicious attacks,

including Heartbleed, Mirai, Torii, and others. The Zeek tool was adapted during the preprocessing phase to extract traffic features. However, a majority of the Mirai attack failed to detect by a Zeek tool, making further analysis of the dataset unfeasible.

As discussed above, numerous network-based datasets are proposed, but creating realistic IoT-based network traffic datasets is still challenging. More precisely, the infrastructure used to simulate attacks in test datasets is unrealistic.

The reliability of any IDS depends on the quality of the data used to generate them. The proposed RT-IoT2022 dataset attempted to fill the informational gap created by preceding datasets in this work. The Table 1 shows the comprehensive analysis of all the state-of-the-art dataset.

Our research contribution includes:

1. The study contributes by employing the Kolmogorov-Smirnov test to validate data distribution assumptions, thereby mitigating bias and ensuring machine learning model's robustness in handling non-normal distributions within network attack datasets.
2. Skewness and kurtosis tests are conducted on the RT-IoT2022 dataset to detect asymmetries in data distribution and potential outliers, providing insights into the dataset's characteristics.
3. PCC and Information Gain Ratio (IGR) techniques are utilized to explore feature correlations, aiding in identifying optimal feature selection and preprocessing strategies to improve model efficiency and predictive accuracy.
4. The analysis is conducted on the recently introduced RT-IoT2022 dataset,[22,23] providing novel insights into its statistical properties and applicability for ML tasks in the context of network attack detection.
5. The validation of test results is performed using both parametric and non-parametric ML models, including SVM with linear and RBF kernels, KNN, Naive Bayes, and Decision trees, ensuring the reliability and generalizability of the findings across different modeling approaches.

Methodology

The proposed infrastructure for creating the RT-IoT2022 dataset of diverse IoT-based network traffic

Table 1 — Dataset comparison of KDDCUP99, Aposemat IoT-23, UNSW-NB15, CICIDS-2017, BoT-IoT, HIKARI-2021, and RT-IoT2022

Dataset	IoT benign network traffic	IoT attack network traffic	Type of benign traffic	Type of attack traffic	Feature types	Feature extraction tool	Practical to generate	Type of attacks
KDDCUP99	No	No	Simulation	Simulation	Packet Based	Bro-IDS tool	No	DoS, Probe, U2R, and R2L
UNSW-NB15	No	No	Real	Real	Packet-based/Flow Based	Argus, Bro-IDS tools	No	Fuzzers Backdoors, DoS, Exploits, Analysis,Generic, Worms Reconnaissance, Shellcode
CICIDS-2017	No	No	Real	Real	Bi-directional Flow Based	CICFlowmeter	No	Brute-Force, DoS/DDoS, Infiltration, BoT, PortScan, Web Attack
BoT-IoT	No	No	Simulation	Real	Packet-based	Argus	Yes	Probing, DoS and Information Theft
HIKARI-2021	No	No	Simulation	Simulation	Bi-directional Flow Based	Zeek	No	Background, Brute Fore, XML, Probing, XMRIGCC
Aposemat IoT-23	Yes	Yes	Real	Real	Bidirectional Flow based	Zeek	Yes	Heartblean, Mirai, Torii
RT-IoT2022 (Proposed)	Yes	Yes	Real	Real	Bi-directional Flow Based	Zeek, Flowmeter	Yes	Brute Force, DDoS, Nmap, ARP Poisoning

is depicted in Fig. 1. This infrastructure comprises the victim network, the attacking network, and routers. The routers are built on Raspberry Pi device and installed with Kali Operating Systems (OS). The attack infrastructure includes two VMs and a Raspberry Pi computer. These attacking devices run Kali, a variant of Linux OS. Penetration testing tools such as Nmap, Wireshark, Ethercap, and Brup suit are included in the Kali OS to establish different forms of cyber attacks. Devices such as Wipro Lights, Thing Speak LEDs, Amazon Echos, and MQTT-Temps are part of the victim network. Utilizing a Raspberry Pi-based router, the attack network communicates with victim networks. The router uses Wire shark, a free and open-source network monitoring tool, to gather information from every network packet. The details of attack traces and benign IoT device traces are shown in Table 2.

Pre-Processing Dataset

The pre-processing phase creates the final RT-IoT2022 datasets of both legitimate and malicious traffic in CSV format as shown in Fig. 2. In the first stage, the Wire shark network analyzer tool installed in the router device capture all generated network traffic using the libpcap library and convert it into Packet Capture (PCAP) files. Attacks such as Distributed Denial of Service (DDoS), ping scans, and spoofing use the weaknesses of standard protocols to exhaust the resources of IoT devices. These cyber attacks are difficult to distinguish from benign network traffic using unidirectional network traces of PCAP files. Therefore, bidirectional features are required for the mitigation of such cyber attacks. Hence, in this research work, we utilized the network analyzer tool Zeek with the "Flowmeter" plugin to extract bidirectional features from CSV files. Flowmeter creates various log files, such as conn.log, flowmeter.log, and mqtt.log, and generates separate logs for each of the services, like DHCP, HTTP, DNS, and so on. Conn.log contains information like IP addresses and listening ports. In conn.log, “uid” is stored together with information like IP addresses and listening ports. Flowmeter.log, on the other hand, contains bi-directional features for each associated “uid”.

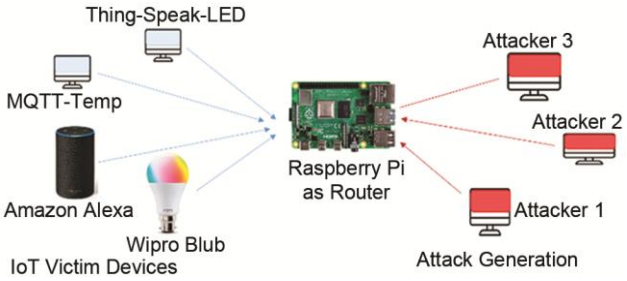

Fig. 1 — Proposed Methodology for RT-IoT2022 Dataset Generation

Statistical Analysis of RT-IoT2022

One of the foundation processes of AI is the transformation of data. The bidirectional network

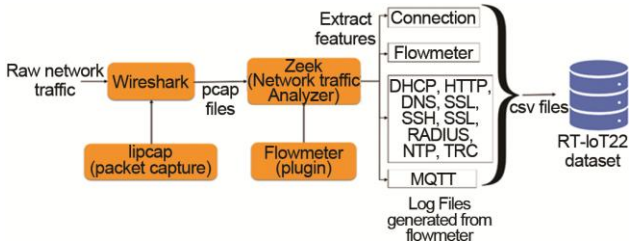

Fig. 2 — Pre-processing of RT-IoT2022 dataset

Table 2 — Details of Attack instances generated in RT-IoT2022 dataset

Category	Subcategory	Tools	Service	Wireshark instances	CIC Flowmeter instances	Class
Normal	ThinkSpeak-LED	—	DNS, HTTP	10526	8108	0
	MQTT-Temp	—	MQTT	8162	4146	0
	Amazon-Alexa	—	DNS, HTTP	6056	5023	0
	Wipro-Bulb	—	SSL, DNS, IRC	1265	253	0
Brute force	SSH Brute Force	Metasploit	SSH, DNS, HTTP	857	37	1
DDoS	SlowlorisDDoS	Slowloris	HTTP, DNS, DHCP	5920	534	2
	SYN Flood DDoS	Hping3	—	712850	94659	3
Nmap	FIN SCAN	Network Mapper	DNS, HTTP	69	28	4
	OS Fingerprinting		—	8008	2000	5
	UDP scan		HTTP, DNS, NTP, DHCP, RADIUS	5606	2590	6
	XMAS Tree scan		DNS, HTTP	8045	2010	7
ARP poisoning	MiTM	Ettercap	DNS, SSL, HTTP, DHCP, NTP	306	7750	8

traffic dataset includes various types of attributes. This results in a significant problem in extracting patterns. An additional problem encounters as the features with a wide range will be emphasized more when we use the original dataset. The independent variables of the original dataset are scaled using feature rescaling techniques to overcome this limitation.

The two main methods for rescaling a dataset are standardization and min-max normalization. The standardization, referred to as the Zeta-score (Z-score), normalizes each variable in the features by transforming it to have a zero mean and unit standard deviation. The formula for standardization is shown in Eq. 1. First, we computed the mean and standard deviation of all the features. Next, we calculated the difference between the mean and each feature. Finally, the resultant value is divided by the standard deviation, as shown in the Eq. 1.

$$Z - score = \frac{x - \mu}{\sigma} \quad \ldots (1)$$

On the other hand, min-max normalization transforms the features into a distribution having a minimum value of 0 and a maximum value of 1 as shown in Eq. 2. The minimum value of each feature is rescaled to 0, while the maximum value is scaled to 1.

$$x_{min\_max} = \frac{x - min(x)}{max(x) - min(x)} \quad \ldots (2)$$

The dataset is initially subjected to the Kolmogorov-Smirnov (KS) test,[24] a widely used statistical technique for the larger dataset. A KS-Test examines whether the dataset corresponds to a normal distribution.

First, we calculate the distance between the Empirical Distribution Function $F_m(x)$ and the Cumulative Distribution function F(x) for all the features.

$$F_m(x) = \begin{cases} \frac{1}{m}\sum_{i=1}^{m} 1 & X_i \le x \\ 0 & otherwise \end{cases} \quad \ldots (3)$$

where, $x \epsilon R$ and m is total number of samples. The KS-Test for F_m is shown in Eq. 4. The score of KS-Test estimates whether an feature (x) rejects the null hypothesis (H_0) of a uniform distribution or fails to reject null hypothesis (alternate hypothesis is H_1) as shown in Eq. 4.

$$KS - Test_{score} = | F_m(x) - F(x) | \quad \ldots (4)$$

The KS-Test score is compared with predefined statistical significance (0.05), and the subsequent score obtained for all features is approximately zero, which is less than statistical significance. Therefore, the attributes in the dataset are non-parametric and nonlinear as the KS-Test rejects the null hypothesis (H_0). The statistical values of the training and testing sets, as shown in Fig. 3, are nearly identical. All values fall within the range of $\tilde{\mu_3}$ 0.28 to 0.52, indicating a good fit.

Further, the categorical features like source, destination port number, protocol and services are converted into numerical forms, which deviate from the statistical probability value.[25] To estimate the asymmetry of the probability distribution of all features, we performed multi-variant skewness analysis on both training and testing sets using the Eq. 5, where is the standardized moment, σ is the standard deviation and μ is the mean. All features are positively skewed, except for source port number, bwd_header_size_min, and bwd_header_size_max, as depicted in Fig. 4. The positively skewed means the asymmetry lies at right hand side of probability distribution. The most skewed features are flow_duration, fwd_iat.min, fwd_iat.tot, bwd_iat.tot and active.tot.

$$\tilde{\mu_3} = \frac{1}{m}\sum_{m}^{i=1}[(\frac{X_i - \mu}{\sigma})^3] \quad \ldots (5)$$

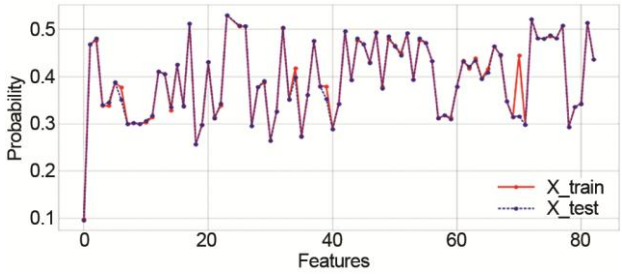

Fig. 3 — KS-test of both training and testing set

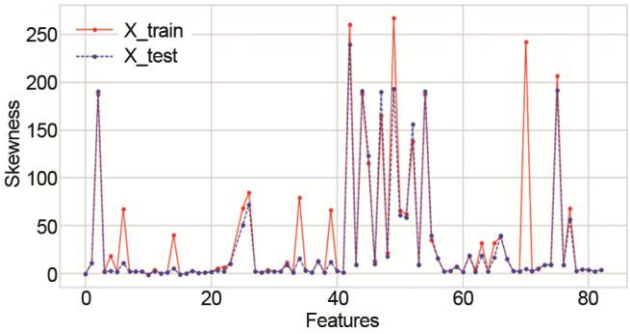

Fig. 4 — Skewness analysis of both training and testing set

On the other hand, multi-variant Kurtosis analysis describes the behavior of the probability distribution tail.[26] It helps to identify the outliers in the dataset. In contrast to normal distributions, distributions with a significant kurtosis have more data in the tail, which gives the impression that features are closer to the mean. The kurtosis (X) function measures the 'tailedness' of all the features using the Eq. 6, where *kurtosis(X)* is the fourth standardized moment, σ is the standard deviation, and μ is the mean. The comparison of *kurtosis(X)* for both the training and testing sets is displayed in Fig. 5. The highest kurtosis features are: 'bwd_iat.tot, fwd_iat.min, active.tot, idle.tot, fwd_iat.tot, active.tot', which is almost the same as highly skewed features.

$$kurtosis(X) = \frac{1}{m}\sum_{i=1}^{m}[(\frac{X_i - \mu}{\sigma})^4] \quad \ldots (6)$$

Feature Selection

In machine learning model, the training of large datasets suffers from computational complexity in terms of the consumption of large system memory and time complexity. Also, the performance degrades if the input variable contains irrelevant and redundant data for the target variable. We investigate this issue by studying the correlation between features using two methods: the PCC and the Information Gain. PCC is the statistical feature selection method that measures the strength of association or correlation between two attributes without considering target values.[27] Correlation coefficients, which can vary from −1 to +1, are used to rank the attributes in order of their correlation. A low correlation is represented by 0, whereas a high correlation is represented by 1. The PCC is estimated using the Eq. 7, where m is size of the dataset, x_i, y_i are data points and μ is the sample mean. The feature-wise correlation coefficients for both the training and test datasets are presented in Fig. 6.

$$PCC_{xy} = \frac{\sum_{i=1}^{m}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{m}(x_i - \mu_x)}\sqrt{\sum_{i=1}^{m}(y_i - \mu_y)}} \quad \ldots (7)$$

The second technique, called Information Gain Ratio (IGR), measures the correlation using class labels for extracting major features.[28] IGR is the ratio of Information Gain (IG) to Intrinsic Values (IV). Consider Y is random variable and x_i is the set of attributes of Y. The formula to measure *IGR* is defined in Eq. 8.

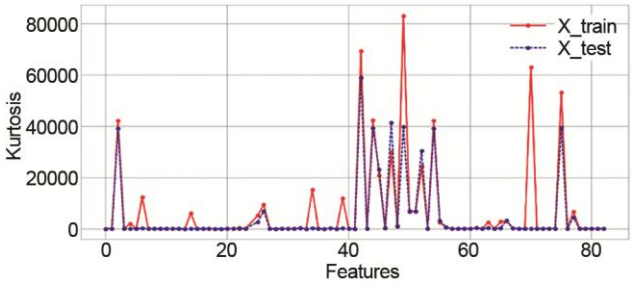

Fig. 5 — Kurtosis analysis of both training and testing set

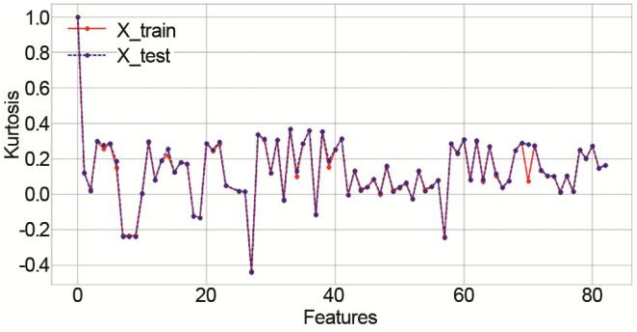

Fig. 6 — PCC of both training and testing set

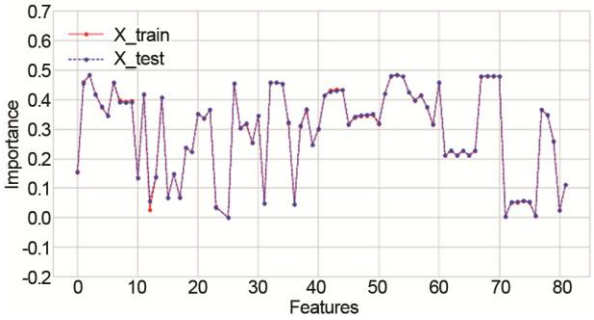

Fig. 7 — IGR of both training and testing set

$$IGR(Y, x_i) = \frac{IG(Y, x_i)}{IV(Y, x_i)} \quad \ldots (8)$$

where, *IG* and *IV* of given observation is estimated using Eq. 9 and 10.

$$IG(Y, x_i) = entropy(Y) - entropy(Y, x_i) \quad \ldots (9)$$

$$IV(Y) = -\sum_{j=1}^{m}\left|\frac{Y_j}{Y}\right| log_2 \left|\frac{Y_j}{Y}\right| \quad \ldots (10)$$

The Information Gain Ratio is plotted for both training and testing set in Fig. 7. The five highest score is attained by the features `flow_iat.max, id.resp_p, active.min, flow_iat.min and active.max'.

Machine Learning Models With RT-IoT2022 Dataset

To evaluate the effectiveness of the dataset, we applied ML models. The observation that the KS-Test

rejects the null hypothesis in the above section confirms that the dataset is non-parametric. As a result of this study, we considered to implement two parametric and three non-parametric machine learning models.[29] They are the Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), and the Decision Tree. The complete dataset is divided into training and testing sets with a ratio of 70% and 30% respectively. Machine learning methods that are not dependent on any assumptions about the mapping function are known as non-parametric models. So, they can learn any functional form from the training data because they make no assumptions.

The training set, weights, and bias of the linear SVM are the sole parameters that make it a parametric ML model. However, the kernel SVM has a kernel function with hyperparameters in addition to the training set, weights, and bias. These hyperparameters grow with the number of training points. Hence SVM with Kernel functions like Radial Bias Function (RBF) and the polynomial is referred to as a non-parametric model.[30] As the number of data points increases, the K also increases.[31] KNN calculates the distances between the vectors using the distance function. Then it predicts the classification based on the majority class among those K points. In this research work, we implemented Euclidean distance as distance function.

Naive Bayes is a probabilistic classifier that utilizes Bayes' theorem along with the assumption of feature independence. It computes the likelihood of a data point belonging to a specific class by considering the probabilities of its features occurring in that class.

Decision Trees (DT) build a tree structure that recursively partitions the data points into child nodes. Using Gini Index or IG criteria, the best feature is selected to split the root node into leaves. The process continues till the tree reaches 100% purity. In this research work, we considered IG as splitting criteria.

Finally, we compared two parametric and three non-parametric benchmark ML models in this study to ensure the reliability of the RT-IoT2022 data.

The KNN algorithm is another non-parametric model used in ML models. KNN interprets K as a hypermeter,

Evaluation Metrics

Model evaluation is an essential aspect of developing a reliable ML model. Building an effective ML model requires careful consideration of the model evaluation.[32] The accuracy, precision, F1-score, and AUC assessment metrics are utilized in this research to quantify the model's performance. The machine learning algorithms were executed on a 64-bit operating system, Windows 10-based, with an x64-based processor. The computational environment boasted 12.0 GB of RAM, ensuring robust performance throughout the experimental process.

The foremost performance metric applied for model evaluation is accuracy, describing the number of correct predictions (True Positive + True Negative) over total predictions. However, accuracy alone may not be sufficient if the dataset has different categories, each containing a different number of instances. Therefore, we consider other parameters, such as F1-score, precision, and recall, to avoid this issue.

For multi-class classification, the precision of a class ‘I’ is the ratio of correctly predicted class ‘I’ to all predictions of class ‘I’. On the other hand, recall is the ratio of correctly predicted class ‘I’ to all actual numbers of class ‘I’. The F1-score metric is also considered in this section to benefit from both precision and recall. To better understand the outcomes when working with an unbalanced dataset, the F1-score measures the harmonic mean of precision and recall.

The ROC curve emphasizes the sensitivity of the ML model.[33] The ROC plot represents the rate of true positives to false positives. The area Under the ROC Curve (AUC) computes the separability. It reveals distinguishing capabilities between different classes. The AUC score of the non-parametric model is higher than that of the parametric models, as shown in Table 3. The graphical representation of the AUC ROC curve is shown in Fig. 8.

Table 3 — Performance evaluation of machine learning models for RT-IoT2022 dataset

Method	Accuracy (%)	Precision	Recall	F1-Score	AUC score	Distribution type	Configurations
SVM - linear	98	0.947	0.701	0.708	0.934	Parametric	Kernel = linear
Gaussian Naïve Bayes	82.5	0.546	0.842	0.552	0.911	Parametric	—
SVM - RBF	98.51	0.976	0.81	0.833	0.948	Non-parametric	Kernel = RBF
KNN	99.74	0.985	0.944	0.961	0.972	Non-parametric	K=2
Decision tree	99.85	0.94	0.9684	0.943	0.984	Non-parametric	Criterion = ginisplitter=best, max_depth = until pure leaves

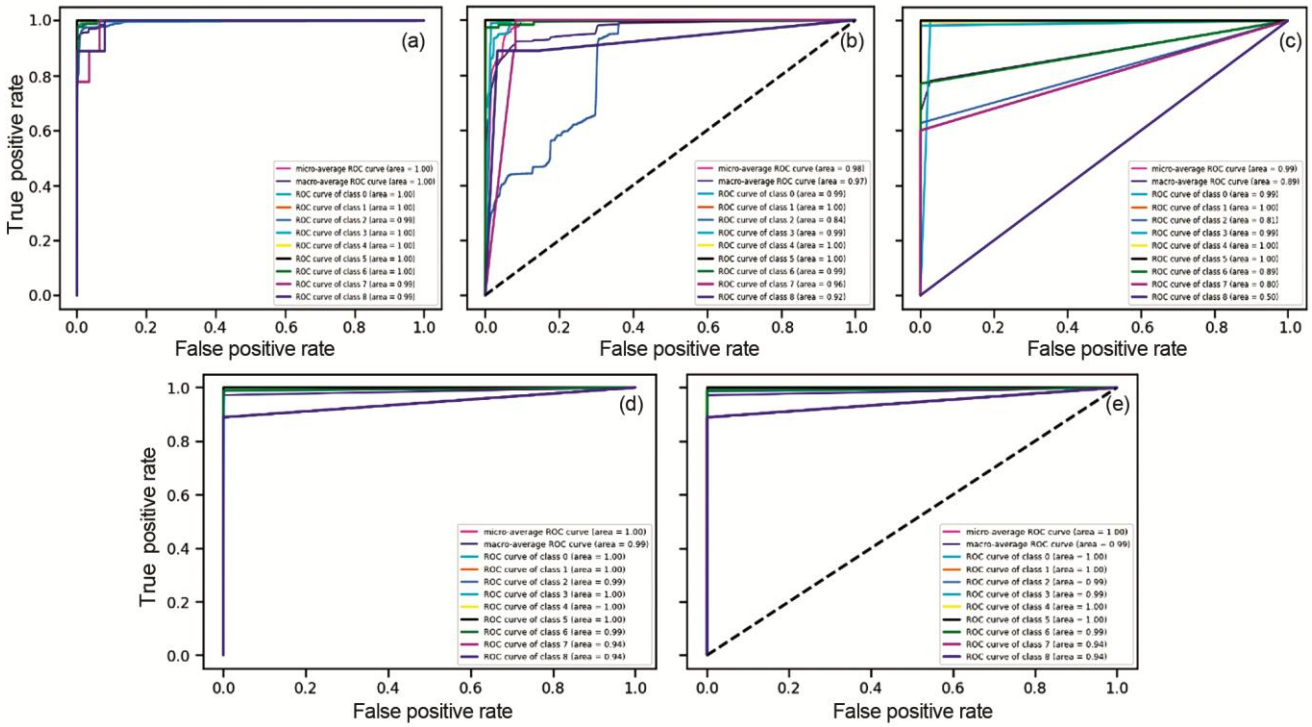

Fig. 8 — AUC ROC curve of machine learning models

In our implementation of SVM, we considered both the parametric (SVM with a “linear” kernel) and non-parametric (SVM with an “RBF" kernel) machine learning models. SVM with the “RBF" kernel performs better than the “linear" kernel in terms of accuracy and precision. Comparing the evaluation metrics of the parametric Naive Bayes algorithm to those of other non-parametric models, such as SVM, KNN, and Decision Tree algorithms reveals a similar decline in effectiveness. Further, the Decision Tree algorithm outperforms by reaching an accuracy of 99.85% with a precision of 0.94%, recall of 0.96%, and F1-Score of 0.943% compared to all other models. The complete analysis is shown in Table 3.

Conclusions

This study addresses the need for IDS systems for IoT devices, leveraging ML techniques on the RT-IoT2022 dataset to identify optimal algorithms.

The statistical test involves KS-Test, Skewness, Kurtosis, PCC and IGR. The KS test indicates the RT-IoT2022 dataset's non-parametric nature, thus favoring non-parametric ML models for better performance. Additionally, skewness and kurtosis tests identified outliers, crucial for preprocessing to ensure dataset balance. The IGR and PCC test demonstrated a good balance between training and testing sets.

Finally, we evaluated both the parametric and non-parametric ML models like SVM (linear, RBF), KNN, Naive Bayes, and Decision Tree. SVM (linear) and Naive Bayes attained the lowest accuracy in the results. In contrast, Decision Tree achieved the highest accuracy of 99.85% among all other models. In future, we aim to develop adaptive models for evolving cyber threats and investigate the privacy preserving techniques to improve detection accuracy while preserving user privacy.

Acknowledgement

We thank The National Institute of Engineering, Mysuru for providing laboratory support to build proprietary dataset.

References

1 Choudhary S & Kesswani N, Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT, *Procedia Comput Sci*, **167** (2020) 1561–1573, https://doi.org/10.1016/j.procs.2020.03.367.

2 Awasthi A & Goel N, Phishing website prediction using base and ensemble classifier techniques with cross-validation, *Cybersecurity*, **5** (2022) 1–23, https://doi.org/10.1186/s42400-022-00126-9.

3 Li Y & Liu Q, A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments, *Energy Rep*, **7(1)** (2021) 8176–8186, https://doi.org/10.1016/j.egyr.2021.08.126.

4 Mehrabi K M, AbuAlhaol I, Raju A Durai, Zhou Y, Giagone R S & Shengqiang H, On building machine learning pipelines for Android malware detection: A procedural survey of practices, challenges and opportunities, *Cybersecurity*, **5(1)** (2022) 1–37, https://doi.org/10.1186/s42400-022-00119-8.

5 Duo R, Nie X, Yang N, Yue C & Wang Y, Anomaly detection and attack classification for train real-time ethernet, *IEEE Access*, **9** (2021) 22528–22541, http://doi.org/10.1109/ACCESS.2021.3055209.

6 Haowen T, An efficient IoT group association and data sharing mechanism in edge computing paradigm, *Cyber Security Appl*, **1** (2023) 100003, https://doi.org/10.1016/j.csa.2022.100003.

7 Morovat K & Panda B, A survey of artificial intelligence in cybersecurity, *Proc2020 Int Conf on Computat Sci Computat Intell* (IEEE), (2020) 109–115, http://doi.org/10.7753/IJCATR1112.1014.

8 Yadav V, Rahul M & Yadav R, A new efficient method for the detection of intrusion in 5g andbeyond networks using machine learning, *J Sci Ind Res*, **80(01)** (2021) 60–65.

9 Das R & Sandhane R, Artificial intelligence in cyber security, *J Physics: Conf Series*, (2021) 1964–42072, http://doi.org/10.1088/1742-6596/1964/4/042072.

10 Couckuyt A, Ruth S, Annelies E, Katrien Q, David N, Sofie V G & Yvan S, Challenges in translational machine learning, *Hum Genet*, **141(9)** (2022) 1451–1466, https://doi.org/10.1007/s00439-022-02439-8.

11 Meintanis S G, Testing for normality with panel data, *J Stat Comput Simul*, **81** (2011) 1745–1752.

12 Subrahmanyam K, Sankar N, S Baggam, S P & Raghavendra R S, A modified KS-test for feature selection, *IOSR J Comput Eng*, **13(3)** (2013) 73–79.

13 Roxas II R, Evangelista M A, Sombillo J A, Nnabuife S G & Pilario K E, Machine learning based flow regime identification using ultrasonic doppler data and feature relevance determination, *Digit Chem Eng*, **3** (2022) 100024, https://doi.org/10.1016/j.dche.2022.100024.

14 Wubiao H, Mingtao D, Zhenhong L, Jianqi Z, Jing Y, Xinlong L, Ling'en M & Yue D, An efficient user-friendly integration tool for landslide susceptibility mapping based on support vector machines: SVM-LSM Toolbox, *Remote Sens*, **14(14)** (2022) 3408, https://doi.org/10.3390/rs14143408.

15 Tavallaee M, Bagheri E, Lu W & Ghorbani A A, A detailed analysis of the KDD CUP 99 data set, *Proc IEEE Symp Computat Intell Secur Defense Appl* (IEEE) 2009, 1–6. https://doi:10.1109/ CISDA.2009.5356528.

16 Moustafa N & Slay J, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), *Proc 2015 Military Commun Informat Syst Conf* (IEEE) 2015, 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

17 Zoghi Z & Serpen G, Unsw-nb15 computer security dataset: Analysis through visualization, (2021), *arXiv Prepr. arXiv2101.0506*.

18 Panigrahi R & Borah S, A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems, *Int J Eng Technol*, **7** (2018) 479–482, https://doi.org/10.14419/ijet.v7i3.24.22797.

19 Koroniotis N, Moustafa N, Sitnikova & Turnbull B, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset, *Fut Gener Comput Syst*, **100** (2019) 779–796, https://doi.org/10.1016/j.future.2019.05.041.

20 Fernandes R & Lopes N, Network intrusion detection packet classification with the HIKARI-2021 dataset: A study on ML Algorithms, *Proc 2022 10th Int Symp Digital Forensic Secur* (IEEE) 2022, 1–5, https://doi.org/10.1109/ISDFS55398.2022.9800807.

21 Parmisano A, Garcia S & Erquiaga M J, Aposemat IoT-23: A labeled dataset with malicious and benign IoT network traffic, *Accessed Jul*, **31** (2020).

22 Sharmila B S & Nagapadma R, QAE-IDS: DDoS anomaly detection in IoT devices using Post-Quantization Training, *Smart Sci*, **11** (2023) 774–789, https://doi.org/10.1080/23080477.2023.2260023.

23 Sharmila B S & Nagapadma R, Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset, *Cybersecurity*, **6(1)** (2023) 41, https://doi.org/10.1186/s42400-023-00178-5.

24 Massey F J, The Kolmogorov-Smirnov test for goodness of fit, *J Am Stat Assoc*, **46** (1951) 68–78.

25 Hubert M & Der Veeken S, Outlier detection for skewed data, *J Chemom A J Chemom Soc*, **22** (2008) 235–246.

26 Livesey J H, Kurtosis provides A good omnibus test for outliers in small samples, *Clin Biochem*, **40** (2007) 1032–1036, https://doi.org/10.1016/j.clinbiochem.2007.04.003.

27 Zihao S, Hui W, Kun L, Peiqian L, Menglong B & Meng Y Z, RP-NBSR: A novel network attack detection model based on machine learning, *Comput Syst Sci Eng*, **37** (2021) 121–133, https://doi.org/10.32604/csse.2021.014988.

28 Sainis N, Srivastava D & Singh R, Feature classification and outlier detection to increased accuracy in intrusion detection system, *Int J Appl Eng Res*, **13(10)** (2018) 7249–7255.

29 Kotlar A M, Iversen B V & de Jong van Lier Q, Evaluation of parametric and nonparametric machine-learning techniques for prediction of saturated and near-saturated hydraulic conductivity, *Vadose Zone J*, **18(1)** (2019) 1–13, https://doi.org/10.2136/vzj2018.07.0141.

30 Patle A & Chouhan D S, SVM kernel functions for classification, in *2013 Int Conf Adv Technol Eng* (IEEE) 2013, 1–9.

31 Li W, Zhang C, Tsung F & Mei Y, Nonparametric monitoring of multivariate data via KNN learning, *Int J Prod Res*, **59** (2021) 6311–6326, https://doi.org/10.1080/00207543.2020.1812750.

32 Sarker I H, Abushark Y B, Alsolami F & Khan A I, Intrudtree: A machine learning based cyber security intrusion detection model, *Symmetry*, **12(5)** (2020) 754, https://doi.org/10.3390/sym12050754.

33 Moualla S, Khorzom K & Jafar A, Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset, *Comput Intell Neurosci*, **2021(1)** (2021) 5557577, https://doi.org/10.1155/2021/5557577.