Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057648

ARTICLE

Backdoor Malware Detection in Industrial IoT Using Machine Learning

Maryam Mahsal Khan[1], Attaullah Buriro[2], Tahir Ahmad[3,*] and Subhan Ullah[4]

[1]Department of Computer Science, CECOS University of IT and Emerging Sciences, Peshawar, 25000, Pakistan

[2]Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Via Torino, Venice, 155, Italy

[3]Center for Cybersecurity, Bruno Kessler Foundation, Trento, 38123, Italy

[4]Faculty of Computer Science, National University of Computer and Emerging Sciences (NUCES-FAST), Islamabad, 44000, Pakistan

*Corresponding Author: Tahir Ahmad. Email: ahmad@fbk.eu

Received: 23 August 2024 Accepted: 19 November 2024 Published: 19 December 2024

ABSTRACT

With the ever-increasing continuous adoption of Industrial Internet of Things (IoT) technologies, security concerns have grown exponentially, especially regarding securing critical infrastructures. This is primarily due to the potential for backdoors to provide unauthorized access, disrupt operations, and compromise sensitive data. Backdoors pose a significant threat to the integrity and security of Industrial IoT setups by exploiting vulnerabilities and bypassing standard authentication processes. Hence its detection becomes of paramount importance. This paper not only investigates the capabilities of Machine Learning (ML) models in identifying backdoor malware but also evaluates the impact of balancing the dataset via resampling techniques, including Synthetic Minority Oversampling Technique (SMOTE), Synthetic Data Vault (SDV), and Conditional Tabular Generative Adversarial Network (CTGAN), and feature reduction such as Pearson correlation coefficient, on the performance of the ML models. Experimental evaluation on the CCCS-CIC-AndMal-2020 dataset demonstrates that the Random Forest (RF) classifier generated an optimal model with 99.98% accuracy when using a balanced dataset created by SMOTE. Additionally, the training and testing time was reduced by approximately 50% when switching from the full feature set to a reduced feature set, without significant performance loss.

KEYWORDS

Industrial IoT; backdoor malware; machine learning; CCCS-CIC-AndMal-2020; security; detection; critical infrastructure

1 Introduction

The rapid proliferation of Industrial Internet of Things (IoT) technologies has introduced various security challenges, making them prime targets for cyber threats and attacks [1,2]. Backdoor malware is a particularly insidious adversary that can infiltrate Industrial IoT setups with stealth and persistence, providing unauthorized access to compromise sensitive data and disrupt operations [3,4]. What makes backdoor malware especially dangerous is its ability to remain dormant and undetectable within the system. Unlike traditional malware, which often aggressively targets and propagates itself, backdoor

malware operates in stealth mode. This stealthy nature of backdoor malware highlights the importance of robust cybersecurity measures. It is crucial to have systems in place that can detect and neutralize these threats before they cause significant damage [5].

Traditional signature-based and behaviour-based backdoor detection methods are inaccurate in detecting sophisticated malware. Signature-based detection methods, while effective against known threats, struggle to identify zero-day exploits. Similarly, behaviour-based detection methods can identify suspicious activities that deviate from the norm, but they may generate a high volume of false positives [6]. In contrast, ML-based approaches have shown promise in detecting backdoor malware [7]. These approaches can adaptively learn from large amounts of data and detect patterns that may indicate a security threat [8]. For instance, some ML methods combine signature-based and behaviour-based features with improving detection accuracy [8]. Other approaches use structured adversarial attacks to determine if a model is backdoored [7]. These machine learning techniques have demonstrated comparatively higher accuracy and usefulness in detecting backdoor malware [9,10]. Hence, by leveraging diverse datasets and designing intelligent algorithms, machine learning approaches can potentially identify and mitigate the subtle signs of backdoor malware, allowing organizations to safeguard their critical infrastructures proactively.

This study focuses on the capabilities of Machine Learning (ML) models in identifying a specific type of malware, known as 'backdoor malware', on the CCCS-CIC-AndMal-2020 dataset. The study investigates the impact of balancing the dataset via resampling techniques, including SMOTE, SDV, and CTGAN, on the performance of the ML models. Further, it evaluates different Machine Learning (ML) models, such as Random Forest (RF), Logistic Regression (LR), Multi-Layer Perceptron (MLP), and AdaBoost (AB), to determine their effectiveness in detecting backdoor malware. The results indicate that RF generated an optimal model with an accuracy of 99.98% when using a balanced dataset generated by SMOTE. Furthermore, the training and testing time were reduced when switching from all features to a reduced feature set. These findings suggest that the use of SMOTE to balance the dataset can significantly improve the performance of ML models in detecting backdoor malware. Additionally, the use of a reduced feature set can help reduce the computational cost of training and testing the models.

In summary, the paper's contributions are as follows:

- The proposal of an ML-based approach to detect backdoor malware in industrial IoT environments. By focusing on Industrial IoT-specific backdoor detection using optimized ML algorithms, we aim to provide a comprehensive and effective solution to safeguard critical infrastructures against backdoor malware attacks.
- Empirical evaluation of different sample augmentation techniques, such as Synthetic Minority Oversampling Technique (SMOTE), Synthetic Data Vault (SDV), and Conditional Tabular Generative Adversarial Network (CTGAN), towards higher accuracy.
- Comparative analysis of computational training and inference time of proposed sample augmentation techniques with and without feature reduction across the different ML models.
- Experimental evaluation of our approach on publicly available CCCS-CIC-AndMal-2020 dataset.

2 Related Work

Backdoor malware is a cybersecurity threat that can bypass security measures to gain unauthorized access to a network, system, or device. This type of malware is particularly dangerous because it allows cybercriminals to steal sensitive data, manipulate operations, and cause extensive

damage to the infrastructure. What makes backdoor malware even more insidious is its ability to remain dormant and undetectable within the system. Traditional detection methods often struggle to identify such sophisticated threats. However, machine-learning approaches have shown promise in detecting backdoor malware. These approaches can adaptively learn from large amounts of data and detect patterns that may indicate a security threat. For instance, some machine learning methods combine signature-based and behavior-based features to improve detection accuracy. Other approaches use structured adversarial attacks to determine if a model is backdoored. These machine-learning techniques have demonstrated comparatively higher accuracy and usefulness in detecting backdoor malware. However, it is important to note that no method is foolproof, and a multilayered defense strategy is often the best approach to cybersecurity.

Research on securing Industrial IoT setups against backdoor malware has been an active area of investigation, driven by growing concerns over the vulnerability of critical infrastructures to cyber threats. This section provides a comprehensive review of the existing literature, focusing on studies related to backdoor malware detection in Industrial IoT environments and the application of machine learning techniques in this domain.

The rapid proliferation of Industrial Internet of Things (IoT) technologies has introduced various security challenges, making these systems prime targets for cyber threats and attacks [1,2]. Researchers have extensively explored the significance of Industrial IoT and the need to address the associated security concerns [11,12]. These studies have laid the groundwork for further investigations into securing Industrial IoT deployments from malicious actors.

In the context of backdoor malware detection in Industrial IoT, existing works have reviewed current approaches and emphasized the crucial importance of robust detection methods to safeguard critical infrastructures [13]. Additionally, surveys on IoT security challenges have proposed machine learning-based solutions for enhancing security, demonstrating the potential of these techniques to address malware-related issues in IoT ecosystems [14]. However, a review of the literature reveals a gap concerning the specific combination of backdoor malware detection and machine learning techniques tailored for Industrial IoT setups [15]. While some studies have explored the application of machine learning for IoT security in general [16,17], limited research has focused on detecting backdoor malware specifically within the Industrial IoT domain [13]. Furthermore, although there is a wealth of research on machine learning for malware analysis in other contexts [15–17], the application of these techniques to Industrial IoT environments using real-world datasets, such as CCCS-CIC-AndMal-2020, has not been thoroughly explored.

To address this research gap, the present study leverages the CCCS-CIC-AndMal-2020 dataset to design and evaluate a machine learning-based detection framework specifically tailored for Industrial IoT scenarios. By focusing on Industrial IoT-specific backdoor detection and integrating optimized machine learning algorithms, this work aims to provide a comprehensive and effective solution to safeguard critical infrastructures against backdoor malware attacks, thereby advancing the state of the art in this domain. Another novel aspect of this work is the exploration of data augmentation techniques, such as SMOTE, SDV, and CTGAN, to balance the imbalanced dataset and improve the performance of the machine learning models.

3 Methodology

The paper presents an ML-based pipeline for effective backdoor malware detection. Fig. 1 depicts our applied methodology. We evaluated our approach on the publicly available CCCS-CIC-AndMal-2020 [18] dataset. This study exploits cleaned, reduced, and balanced data samples. The methodology is further explained in detail below.

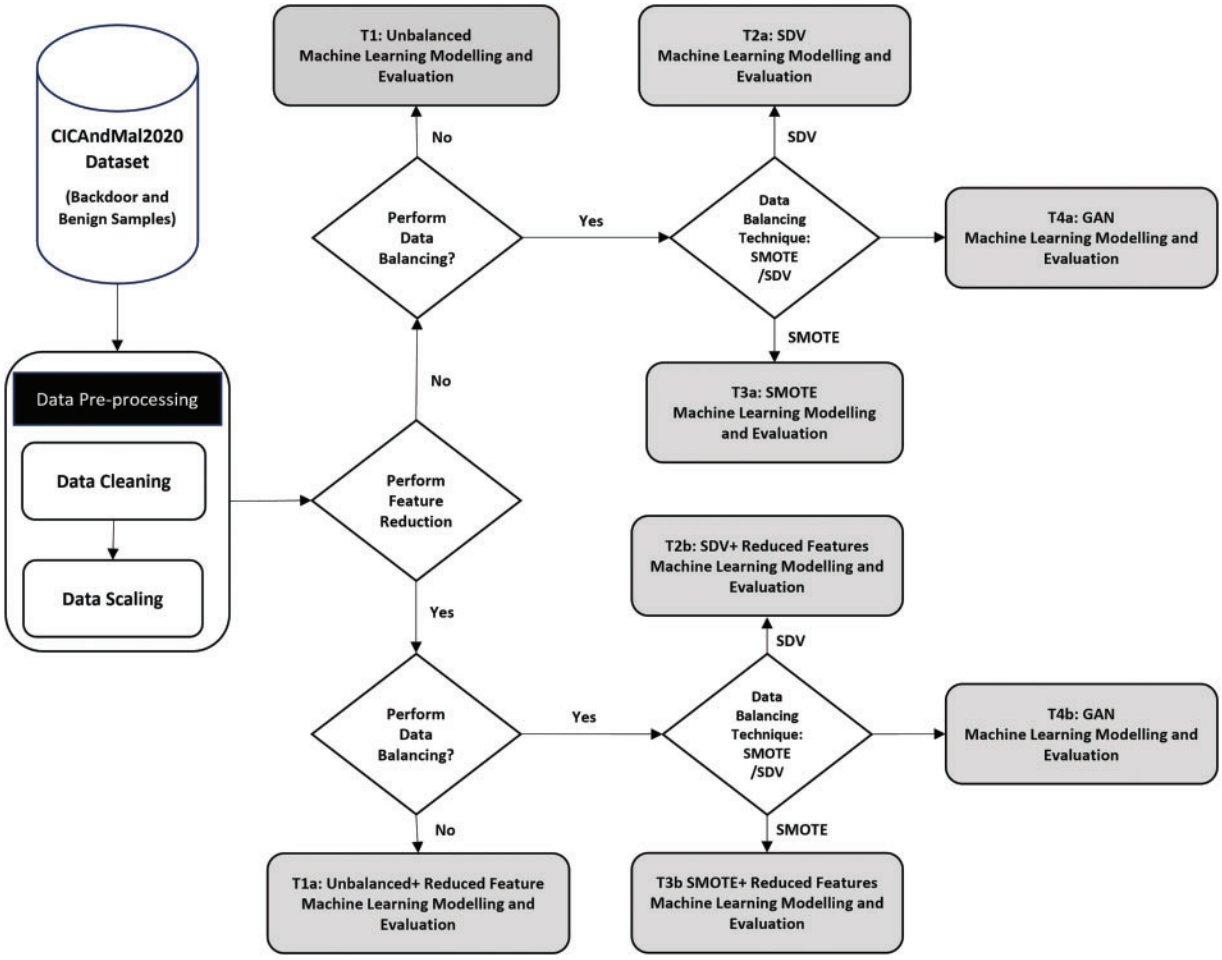

Figure 1: Machine learning model generation across three different strategies of balancing data samples representation, including training across full feature set and reduced feature set

3.1 Backdoor Malware Dataset

We use a publicly available dataset provided by the Canadian Centre for Cyber Security and Canadian Institute for Cybersecurity (CCCS-CIC) collaboration project CCCS-CIC-AndMal-2020 [18]. The dataset consists of 14 malware categories, including Adware, Backdoors, Trojan-Banker, No Category, Trojan-Dropper, File-Infector, PUA, Ransomware, Riskware, Scareware, Trojan-SMS, Trojan-SPY, and ZeroDay. Features of both static and dynamic analysis are provided in the dataset. The current study focuses on the static analysis of malware features. The static analysis file consists of four Benign Comma-Separated Values (CSV) files, i.e., Ben0.csv, Ben1.csv, Ben2.csv, and Ben3.csv.

The total number of benign data samples is 130,427. On the other hand, Backdoor.csv has a total of 1538 samples. Static features that are mapped across 9504 samples include (1) Activities that reflect on one screen of the Android app's user interface, (2) Broadcast receivers and providers, (3) Permission requests, i.e, to protect the privacy of the user and is needed to access sensitive user data, (4) System features such as camera or internet and (5) Metadata information. More information can be found in [19,20]. Imbalance datasets are found to negatively impact the performance and reliability of ML models [21,22]. Ways to mitigate this issue include resampling and selecting algorithms that are less sensitive to imbalanced datasets or cost-sensitive learning [21]. The current study shall focus on balancing the dataset through oversampling. The oversampling method involves creating synthetic examples of the minority classes. It helps to prevent the loss of information in the majority caused by undersampling but can result in a risk of overfitting if not handled properly. Three oversampling strategies shall be used and include synthetic minority over-sampling technique (SMOTE) [23] method, Conditional Tabular Generative Adversarial Network (CTGAN) [24] and Synthetic Data Vault (SDV) [25]. More details ahead.

3.2 Feature Reduction

Several techniques are commonly employed for feature reduction such as mutual information selection [26], recursive feature elimination [27], and Pearson correlation coefficient [28]. For a quick exploratory analysis, Pearson Correlation Coefficient (PCC) is a popular technique used in the realm of machine learning for feature reduction including many android malware detection methods [28–30]. It evaluates the similarity and strength of a linear relationship between two continuous variables (features). The predictive performance and computational speed of the ML model improve by dropping strongly correlated features. The dataset investigated in the current study consists of 9504 features. PCC of the features are computed. Features with an absolute correlation coefficient of 0.9 and higher are considered highly correlated features and are eliminated from the dataset, resulting in 2620 uncorrelated features. Setting a lower criterion would result in more than 90% feature deletion, therefore the threshold was set at 0.9.

3.3 Data Augmentation

In real-world datasets, including the dataset used in the current study, there are more benign files than malicious ones making the dataset imbalanced. Data imbalance is a common challenge in malware detection, affecting the model's ability to detect rare malware samples effectively [31,32]. Malware samples are harder to collect as they may appear in isolated instances or target specific environments [33]. Machine learning models trained on imbalanced datasets often become biased towards predicting the majority class, i.e., the model will likely predict more files as benign and miss the rare malware samples (false negatives), overfitting to the majority class [32]. Data augmentation techniques such as oversampling [23], cost-sensitive based learning [34], and using ensemble methods such as bagging and boosting [35] can help to reduce problems associated with an imbalance dataset. Many of the data augmentation techniques introduce artificial samples by generating new data points from existing data [36]. Data augmentation techniques are needed for several reasons: Firstly, they help prevent models from overfitting, especially when the initial training set is too small. Secondly, they improve model accuracy by increasing the diversity and size of the training set. Finally, they reduce the operational cost of labeling and cleaning the raw dataset. However, the application of data augmentation techniques to tabular data is a relatively new area of research [37,38], and finding an effective approach has always been extremely challenging. As such, we exploit multiple data augmentation schemes explained below.

3.3.1 Synthetic Minority Oversampling Technique

The Synthetic Minority Oversampling Technique, or SMOTE, is the simplest yet powerful data augmentation method used to address the issue of class imbalance problem [23,36]. It works by generating synthetic examples for the minority class, thereby balancing the class distribution. SMOTE is simple yet useful because it does not merely duplicate existing minority samples but rather creates new synthetic instances that are plausible and close to the feature space of the minority class [39]. SMOTE differs from other approaches in that it does not rely on random oversampling or undersampling, which can lead to overfitting or loss of potentially useful data, respectively [39]. Instead, SMOTE generates synthetic examples by interpolating between existing minority instances, thereby enriching the dataset with new, informative examples. SMOTE has been widely adopted in numerous studies dealing with imbalanced datasets, particularly in the context of tabular data. Its popularity stems from its effectiveness in improving the performance of various machine learning models, making it a go-to technique for many researchers in the field including malware detection [36,39–41].

3.3.2 Synthetic Data Vault

The Synthetic Data Vault (SDV) [25] is a Python library designed for synthesizing data. It can mimic data in a table, across multiple relational tables, or time series. SDV is simple and useful because it provides a practical solution to common challenges such as limited data and overfitting. The SDV contains multiple models, ranging from classical statistical methods (GaussianCopula) to deep learning methods (CTGAN), to generate data for single tables, multiple connected tables, sequential tables, etc. SDV differs from other approaches in its ability to handle different types of data structures, including tabular data, time-series data, and multi-table data. This makes it a versatile tool for data augmentation across a wide range of applications including malware detection [42].

3.3.3 Conditional Tabular Generative Adversarial Network

Conditional Tabular Generative Adversarial Network, or CTGAN [43], is a deep learning-based synthetic data generator specifically designed for single table data by SDV creators [25]. It is simple and useful because it can learn from real data and generate synthetic data with high fidelity and has been effectively used by researchers in the field of malware detection [44,45]. This makes CTGAN a valuable tool for tasks such as data augmentation, especially when the available real data is limited.

3.4 Model Development

In this paper, we use six simple yet state-of-the-art classifiers to detect backdoor malware on the CICAndMal2020 dataset. The classifiers are Random Forest (RF), Logistic Regression (LR), AdaBoost (AB), Perceptron (PER), Deep Neural Network (DNN), and Multi-Layer Perceptron (MLP); we choose these classifiers based on the following criteria:

- **Scalability:** We exploited these classifiers because they can handle large and high-dimensional data efficiently and effectively. RF, LR, and AB are scalable classifiers that can deal with imbalanced and noisy data. MLP and DNN are also scalable, however, they require more computational resources and tuning than the others.
- **Interpretability:** We chose these classifiers because they provide meaningful and transparent results that can be explained and understood by domain experts. RF and LR are interpretable classifiers that can provide feature importance and decision rules. AB, MLP, and DNN are

less interpretable, but they can offer insights into the data distribution and the nonlinear relationships between features and classes.

- **Performance:** We chose these classifiers because they have been shown to provide higher accuracy and robustness in detecting and classifying backdoor malware. RF, AB, MLP, and DNN are powerful classifiers that can capture complex and nonlinear patterns in the data. LR, however, is a simpler classifier, but it can perform well on linearly separable data and provide a baseline for comparison.

After feature reduction, for both balanced and imbalanced representations, the dataset is split into two sets: one for training and one for testing. A stratified 5-fold cross-validation strategy is used on the training set to train the ML/DL models, while the testing set is used to report the average testing accuracy of the trained models. Standard performance metrics for evaluating supervised algorithms, discussed in Section 3.5, are computed and reported in Fig. 1 for both balanced and imbalanced data representations, respectively. All these steps are carried out in the development environment with Intel Core i7 7820HQ-processor, 32 GB DDR4 RAM, and Windows 10 operating system.

3.5 Performance Metrics

Several performance metrics are commonly used to evaluate a model's performance in machine learning classification problems. These metrics include accuracy, precision, recall, and F1-score, each measuring different aspects of classification performance, Where the F1-score is a good measure, particularly in cases where the dataset is imbalanced [46].

- **Accuracy:** Accuracy measures how accurately a classification model is applied overall. It determines the proportion of accurately predicted occurrences to all of the dataset's instances and is mathematically computed using Eq. (1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  where
  - TP (True Positives) is the number of correctly predicted positive instances.
  - TN (True Negatives) is the number of correctly predicted negative instances.
  - FP (False Positives) is the number of instances that were actually negative but were incorrectly predicted as positive.
  - FN (False Negatives) is the number of instances that were positive but were incorrectly predicted as negative.
- **Precision:** Precision measures the accuracy of positive predictions made by the model. It calculates the ratio of true positives to the total number of positive predictions expressed in Eq. (2).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- **Recall:** Recall measures the ability of the model to correctly identify positive instances. It calculates the ratio of true positives to the total number of actual positive instances, expressed in Eq. (3).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- **F1-score:** The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when dealing with imbalanced datasets, expressed in Eq. (4).

$$\text{F1score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

- **Confusion Matrix:** The confusion matrix is a table that summarizes a machine learning model's performance on a set of test data. It shows the number of true positive, true negative, false positive, and false negative predictions that the model made for each class.

4 Results and Discussion

Table 1 presents the results of experiments using six machine learning models: RF, LR, AB, PER, MLP, and DNN. The model's performance is compared to that of the original imbalanced dataset and the balanced datasets using each of the three strategies. The performance of the models is also compared to the full feature set and the reduced feature set. Four metrics are used to measure the performance of the models: accuracy, precision, recall, and F1-score. The F1-score for an unbalanced dataset across full and reduced feature sets was approximately 96%. With a balanced dataset, across the three strategies SMOTE, CTGAN, and SDV, the F1-score is found to be greater than 99.95%. This suggests that balancing the dataset has the potential to improve the generalization of the ML model by reducing overfitting.

RF is one of the most powerful ensemble classifiers often used in machine learning applications. It has been found successful on many benchmarked datasets. In Table 1, we see that all six classifiers performed well across all tasks, with only slight variations in metrics. However, RF outperforms other classifiers in terms of accuracy and F1-score under the No Balance task with original features, and it maintains its high performance when reduced features are employed.

Table 1: Performance of machine learning algorithms on a balanced representation of the CCCS-CIC-AndMal-2020 dataset. The current study investigates the performance of the model on a balanced dataset achieved through oversampling techniques such as SMOTE, CTGAN and SDV. Results when no balancing technique is also shown here as a baseline (for reference). Highest numbers are highlighted in bold

Datasets	Classification results of trained classifiers									
Tasks	Original features					Reduced features				
	Models	Accr	Pre	Rec	F1	Models	Accr	Pre	Rec	F1
No balance	RF	**0.9983**	0.9832	0.9446	0.9631	RF	**0.9982**	0.9830	0.9404	0.9607
	AB	0.9967	0.9612	0.8924	0.9240	AB	0.9962	0.959	0.8712	0.9103
	LR	0.9935	0.8381	0.9264	0.8770	LR	0.9963	0.9180	0.9268	0.9224
	PER	0.9894	0.7665	0.9317	0.8288	PER	0.9928	0.832	0.8904	0.8588
	MLP	0.9973	0.9407	0.9462	0.9434	MLP	0.9975	0.945	0.9516	0.9483
	DNN	0.9941	0.9418	0.9947	0.9666	DNN	0.9973	0.9777	0.9911	0.9843
	Models	Accr	Pre	Rec	F1	Models	Accr	Pre	Rec	F1

(Continued)

Table 1 (continued)

Datasets	Classification results of trained classifiers									
Tasks	Original features					Reduced features				
	Models	Accr	Pre	Rec	F1	Models	Accr	Pre	Rec	F1
SMOTE	RF	**0.9998**	0.9998	0.9998	0.9998	RF	**0.9997**	0.9997	0.9997	0.9997
	AB	0.9995	0.9995	0.9995	0.9995	AB	0.9995	0.9995	0.9995	0.9995
	LR	0.9988	0.9987	0.9988	0.9988	LR	0.9987	0.9987	0.9987	0.9987
	PER	0.9908	0.9909	0.9909	0.9908	PER	0.9936	0.9936	0.9937	0.9936
	MLP	0.9988	0.9988	0.9988	0.9988	MLP	0.9988	0.9987	0.9988	0.9988
	DNN	0.9988	0.9987	0.9988	0.9988	DNN	0.999	0.999	0.999	0.999
	Models	Accr	Pre	Rec	F1	Models	Accr	Pre	Rec	F1
SDV	RF	**0.9996**	0.9996	0.9996	0.9996	RF	**0.9997**	0.9997	0.9997	0.9997
	AB	0.9991	0.9991	0.9991	0.9991	AB	0.9995	0.9995	0.9995	0.9995
	LR	0.9951	0.9952	0.995	0.9951	LR	0.9963	0.9964	0.9963	0.9963
	PER	0.989	0.9893	0.9889	0.989	PER	0.9914	0.9916	0.9914	0.9914
	MLP	0.9981	0.9981	0.9981	0.9981	MLP	0.9978	0.9978	0.9978	0.9978
	DNN	0.9976	0.9976	0.9976	0.9976	DNN	0.9981	0.9981	0.9981	0.9981
	Models	Accr	Pre	Rec	F1	Models	Accr	Pre	Rec	F1
CTGAN	RF	**0.9996**	0.9996	0.9996	0.9996	RF	**0.9997**	0.9997	0.9997	0.9997
	AB	0.9995	0.9995	0.9995	0.9995	AB	0.9995	0.9995	0.9995	0.9995
	LR	0.9794	0.9798	0.9792	0.9794	LR	0.9794	0.9798	0.9792	0.9794
	PER	0.9793	0.9796	0.9791	0.9793	PER	0.8848	0.8871	0.8844	0.8845
	MLP	0.9980	0.9980	0.9980	0.9980	MLP	0.9963	0.9963	0.9963	0.9963
	DNN	0.9976	0.9976	0.9976	0.9976	DNN	0.9971	0.9971	0.9971	0.9971

RF generates feature significance by computing the mean decrease in impurity or Gini impurity, which is associated with an increase in feature weights. Fig. 2a,b shows the feature significance graph, in descending order, for the three balanced dataset representation (SMOTE, SDV, CTGAN) using all features Fig. 2a and reduced feature set Fig. 2b, respectively. It is observed that the common features in the top 10 significant features include feature index 12 and 9134, while in the reduced feature, the common features in the top 10 were found to be feature index 11 and 2501. Moreover, out of the 9504 features, approximately 16.31%–17% of the features are used in the RF models, while in the reduced space, out of the 2620 features, approximately 35.5%–37% features are used in the generated RF models. This indicates the presence of redundant features, whereby their removal can be used to further improve computational performances.

Fig. 3a displays the time (in seconds) taken to train. In contrast, Fig. 3b shows the testing time (in seconds) for the **RF** models across three balanced data representations (BS_SMOTE, BS_SDV, BS_CTGAN) with all and reduced features, respectively. The training and testing times are found to reduce by approximately 50% when a reduced feature set is used. The reduction in training time is also found to incrementally improve the accuracy of the model by 0.01%.

Table 2 shows the confusion matrix for the **RF** models generated using SMOTE, SDV, and CTGAN data representations with all and reduced feature sets on the testing set. It is observed with

a reduced feature set the false negatives (Backdoor as Benign) samples are further reduced for the balanced data representation in SDV and CTGAN. Further analysis on how to minimize undetected threats via other feature elimination strategies will be explored in the future. Moreover, it is observed that apart from SMOTE, incremental improvement in the prediction of backdoor samples in SDV and CTGAN is obtained.

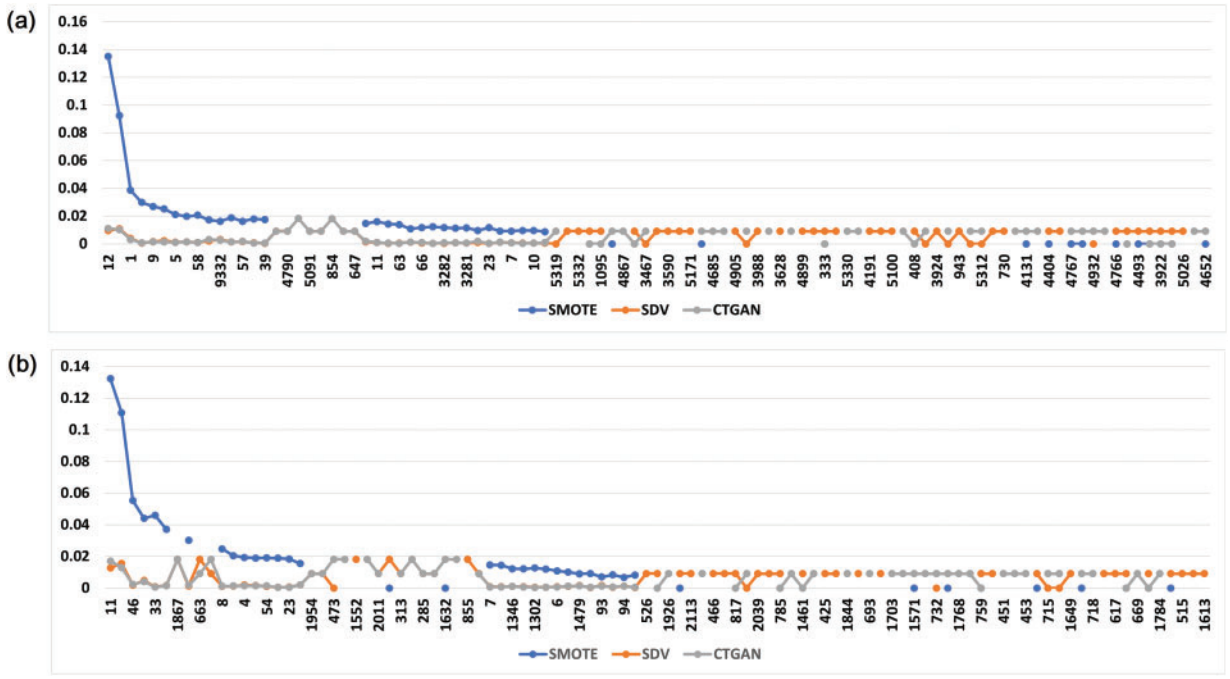

Figure 2: Feature significance graph, in descending order, extracted from the best performing RF models on three balanced dataset representations: SMOTE, SDV, and CTGAN using (a) all features, and (b) reduced feature set, respectively

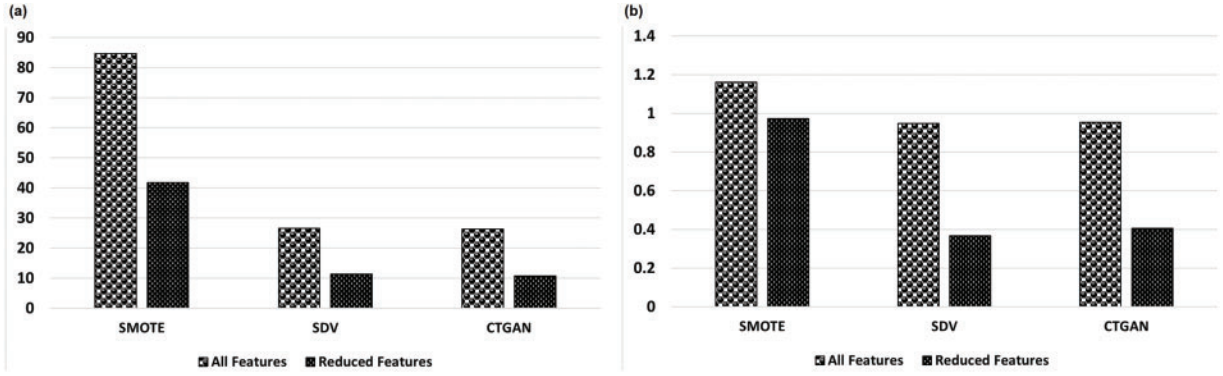

Figure 3: Average time taken, in seconds, to (a) train and (b) test RF models on three balanced representations of data samples, i.e., SMOTE, SDV, and CTGAN across all and reduced feature space

The results presented in this study demonstrate the effectiveness of machine learning models, particularly RF, in detecting backdoor malware in Industrial IoT environments. The high accuracy

(99.98%) and F1-score achieved by the RF model when using the SMOTE-balanced dataset highlight the potential of our approach to reliably identify backdoor threats in Industrial IoT setups. Furthermore, the analysis of feature significance reveals that only a subset of the full feature set (16%–17%) and reduced feature set (35%–37%) are utilized by the RF models. This suggests the presence of redundant features, which can be removed to optimize the computational efficiency of the detection framework without significantly impacting the overall performance.

Table 2: Confusion Matrices of the RF models for three balanced representations, i.e., SMOTE, SDV, and CTGAN, across full and reduced feature sets

Datasets	Features				
Tasks		Original features		Reduced features	
SMOTE	Actual/Predict	*Benign*	*Backdoor*	*Benign*	*Backdoor*
	Benign	9737	3	9737	3
	Backdoor	0	9511	2	9509
SDV	Actual/Predict	*Benign*	*Backdoor*	*Benign*	*Backdoor*
	Benign	9757	4	9757	4
	Backdoor	10	9953	8	9955
CTGAN	Actual/Predict	*Benign*	*Backdoor*	*Benign*	*Backdoor*
	Benign	9757	4	9757	4
	Backdoor	7	9956	6	9957

Since backdoor malware typically represents a small fraction of the overall dataset, the isolation forest [47], an unsupervised learning algorithm, is particularly effective because it doesn't require large amounts of labeled data for training. Instead, it detects outliers by assigning scores based on the number of partitions needed to isolate a particular data point. Points with high anomaly scores are flagged as potential backdoor malware, facilitating timely detection and response. Isolation forest, with the parameters {n_estimators = 100, max_samples = 15, contamination = 0.5}, was applied on a balanced backdoor dataset in the study achieving a testing accuracy was 79%. Further research on the potential of unsupervised learning algorithms in backdoor malware detection will be conducted in the future.

Feature reduction techniques, such as Recursive Feature Elimination (RFE), are valuable in optimizing model performance by identifying the most relevant features in a dataset, which is particularly useful when dealing with high-dimensional data like malware detection datasets. In the current study, RFE is applied to the feature reduced balanced dataset consisting of 2620 features, to iteratively eliminate the least important features up to 1000 features, refining the dataset to contain only those features that contribute most significantly to classification accuracy. After feature reduction, a Random Forest classifier is employed to classify the samples into malicious or benign categories, which generated an accuracy of 100%. Through this experiment it is clear that further reducing dataset's dimensionality has the potential not only improve the computational time but also accurate detection of malicious backdoor samples.

Compared to traditional signature-based and behaviour-based detection methods, our machine learning-based approach offers several advantages: Unlike the rigid, rule-based nature of signature-based detection, our models can adaptively learn from data and identify more sophisticated, previously unseen patterns of backdoor malware. Additionally, the fact that ML models demonstrate higher accuracy and lower false positive rates compared to behaviour-based detection-which can struggle with high-dimensional, complex Industrial IoT data, our approach could be handy. Further. in terms of computational efficiency, the 50% reduction in training and testing time when using the reduced feature set highlights the potential for our approach to be deployed in resource-constrained Industrial IoT environments. This optimization of computational cost is crucial for enabling real-time, scalable detection of backdoor threats without compromising the overall performance.

Overall, the findings of this study provide valuable insights into the development of effective backdoor malware detection systems for Industrial IoT. By leveraging RF, optimizing the feature set, and smote augmentation technique, our proposed approach offers a comprehensive and efficient solution to safeguard critical infrastructures against this emerging cyber threat.

5 Conclusions

Backdoor malware detection is a crucial aspect of Industrial IoT security, as it can help prevent unauthorized access to sensitive data and systems. The key contribution of this study is the development of a novel machine learning-based approach for effectively detecting backdoor malware in Industrial IoT environments. By leveraging the CCCS-CIC-AndMal-2020 dataset, which provides real-world scenarios of backdoor malware in Industrial IoT systems, we have designed and evaluated a detection framework that combines feature engineering, sample augmentation, and classification modules. The results demonstrate the superior performance of the Random Forest (RF) classifier, which achieved an accuracy of 99.98% when using a balanced dataset generated by the SMOTE technique. This highlights the effectiveness of our approach in reliably identifying hidden backdoors and enhancing the security of Industrial IoT deployments. Furthermore, the study reveals that only a subset of the full feature set (16%–17%) and reduced feature set (35%–37%) are utilized by the RF models. This finding suggests the presence of redundant features, which can be removed to optimize the computational efficiency of the detection framework without significantly impacting the overall performance. The 50% reduction in training and testing time when using the reduced feature set underscores the potential for our approach to be deployed in resource-constrained Industrial IoT environments.

Future research directions could focus on exploring the performance of various machine and deep learning models including gradient boosting machines, across different feature elimination strategies such as recursive elimination, and mutual information, to reduce undetected threats further and explore other mechanisms of balancing the malware dataset.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Maryam Mahsal Khan, Attaullah Buriro; data collection: Subhan Ullah, Maryam Mahsal Khan; analysis and interpretation of results: Tahir Ahmad, Attaullah Buriro, Subhan Ullah; draft manuscript preparation: Tahir Ahmad, Maryam Mahsal Khan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] J. M. Mcginthy and A. J. Michaels, "Secure industrial internet of things critical infrastructure node design," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8021–8037, 2019. doi: 10.1109/JIOT.2019.2903242.

[2] A. A. Mirani, G. Velasco-Hernandez, A. Awasthi, and J. Walsh, "Key challenges and emerging technologies in industrial iot architectures: A review," *Sensors*, vol. 22, no. 15, 2022, Art. no. 5836. doi: 10.3390/s22155836.

[3] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of things," *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 2985–2996, 2020. doi: 10.1109/TII.2020.3023507.

[4] L. P. Ledwaba and G. P. Hancke, "Security challenges for industrial IoT," in *Wireless Networks and Industrial IoT: Applications, Challenges and Enablers*. Cham: Springer, 2021, pp. 193–206.

[5] P. Victor, A. H. Lashkari, R. Lu, T. Sasi, P. Xiong and S. Iqbal, "IoT malware: An attribute-based taxonomy, detection mechanisms and challenges," *Peer-to-Peer Netw. Appl.*, vol. 16, no. 3, pp. 1–52, 2023. doi: 10.1007/s12083-023-01478-w.

[6] Y. Xu, X. Liu, K. Ding, and B. Xin, "IBD: An interpretable backdoor-detection method via multivariate interactions," *Sensors*, vol. 22, no. 22, 2022, Art. no. 8697. doi: 10.3390/s22228697.

[7] C. Yang, "Detecting backdoored neural networks with structured adversarial attacks," 2021. Accessed: Sep. 10, 2024. [Online]. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-90.pdf

[8] H. -Y. Kwon, T. Kim, and M. -K. Lee, "Advanced intrusion detection combining signature-based and behavior-based detection methods," *Electronics*, vol. 11, no. 6, 2022, Art. no. 867. doi: 10.3390/electronics11060867.

[9] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan and S. Chattopadhyay, "Model agnostic defence against backdoor attacks in machine learning," *IEEE Trans. Reliab.*, vol. 71, no. 2, pp. 880–895, 2022. doi: 10.1109/TR.2022.3159784.

[10] H. Fu, A. Sarmadi, P. Krishnamurthy, S. Garg, and F. Khorrami, "Mitigating backdoor attacks on deep neural networks," in *Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Use Cases and Emerging Challenges*. Cham: Springer, 2023, pp. 395–431.

[11] M. Aazam, S. Zeadally, K. A. Harras, and E. Dutkiewicz, "Industrial internet of things: Challenges, opportunities, and directions," *Future Gener. Comput. Syst.*, vol. 81, pp. 1–3, 2018.

[12] Y. A. Almohri, M. Alshehri, S. Alshahrani, M. Alomari, and O. S. Albahri, "Security analysis of industrial internet of things: A review," *IEEE Access*, vol. 7, pp. 102 109–102 122, 2019.

[13] I. Bisio, A. Gaglione, M. Marchetti, and F. Lavagetto, "Detecting malware in industrial IoT networks: A review," *Sensors*, vol. 21, no. 6, 2021, Art. no. 2038.

[14] M. S. Siddiqui, E. Sabir, and S. Shamim, "IoT security challenges and machine learning-based solutions: A survey," *Comput. Secur.*, vol. 91, 2020, Art. no. 101718.

[15] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, no. 2, 2012, Art. no. 6. doi: 10.1145/2089125.2089126.

[16] A. Ferrante, S. Ognawala, A. Lanzi, and D. Balzarotti, "Scalable, behavior-based malware clustering," in *Proc. 2013 IEEE Symp. Secur. Priv. (SP)*, 2013, pp. 25–39.

[17] B. Kolosnjaji, A. V. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Proc. 14th Int. Conf. Mach. Learn. Data Mini. Pattern Recognit. (MLDM)*, 2018, pp. 439–453.

[18] Kaggle, "CCCS-CIC-AndMal-2020 dataset," 2020. Accessed: Sep. 10, 2024. [Online]. Available: https://www.kaggle.com/ahmedmohamedashraf/cic-andmal2020
[19] D. S. Keyes, B. Li, G. Kaur, A. H. Lashkari, F. Gagnon and F. Massicotte, "EntropLyzer: Android malware classification and characterization using entropy analysis of dynamic characteristics," in *2021 Reconcil. Data Analyt., Automat., Priv. Secur.: A Big Data Challen. (RDAAPS)*, 2021, pp. 1–12.
[20] A. Rahali, A. H. Lashkari, G. Kaur, L. Taheri, F. Gagnon and F. Massicotte, "DIDroid: Android malware classification and characterization using deep image learning," in *Proc. 2020 10th Int. Conf. Commun. Netw. Secur., ICCNS '20*, New York, NY, USA, Association for Computing Machinery, 2021, pp. 70–82. doi: 10.1145/3442520.3442522.
[21] S. B. Kotsiantis, D. N. Kanellopoulos, and P. E. Pintelas, "Handling imbalanced datasets: A review," 2006. Accessed: Sep. 10, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:14354369
[22] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets-a review paper," in *Proc. 16th Midwest Artif. Intell. Cognit. Sci. Conf.*, 2005.
[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002. doi: 10.1613/jair.953.
[24] I. J. Goodfellow *et al.*, "Generative adversarial networks," 2014, *arXiv:1406.2661*.
[25] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 399–410.
[26] M. Gazzan and F. T. Sheldon, "An incremental mutual information-selection technique for early ransomware detection," *Information*, vol. 15, no. 4, 2024, Art. no. 194. doi: 10.3390/info15040194.
[27] B. Sani Mahmoud, B. Ahmad, and A. Garko, "A machine learning model for malware detection using recursive feature elimination (RFE) for feature selection and ensemble technique," *IOSR J. Comput. Eng.*, vol. 24, no. 8, pp. 23–30, 2022.
[28] P. Kishore, S. K. Barisal, and D. Sani Mahmoud, "Javascript malware behaviour analysis and detection using sandbox assisted ensemble model," in *2020 IEEE Reg. 10 Conf. (TENCON)*, 2020, pp. 864–869.
[29] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han and X. Zhang, "Exploring permission-induced risk in android applications for malicious application detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 11, pp. 1869–1882, 2014. doi: 10.1109/TIFS.2014.2353996.
[30] X. Wang and C. Li, "Android malware detection through machine learning on kernel task structures," *Neurocomputing*, vol. 435, no. 6, pp. 126–150, 2021. doi: 10.1016/j.neucom.2020.12.088.
[31] Z. Sawadogo, G. Mendy, J. M. Dembele, and S. Ouya, "Android malware detection: Investigating the impact of imbalanced data-sets on the performance of machine learning models," in *2022 24th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2022, pp. 435–441.
[32] R. Oak, M. Du, D. Yan, H. Takawale, and I. Amit, "Malware detection on highly imbalanced data through sequence modeling," in *Proc. 12th ACM Workshop Artif. Intell. Secur., AISec'19*, New York, NY, USA, Association for Computing Machinery, 2019, pp. 37–48. doi: 10.1145/3338501.3357374.
[33] T. Li *et al.*, "A malware detection model based on imbalanced heterogeneous graph embeddings," *Expert. Syst. Appl.*, vol. 246, no. 27, 2024, Art. no. 123109. doi: 10.1016/j.eswa.2023.123109.
[34] X. Hu *et al.*, "Cost sensitive gnn-based imbalanced learning for mobile social network fraud detection," 2023, *arXiv:2303.17486*.
[35] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin and N. N. Abdullah, "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," in *Proc. First Int. Conf. Adv. Data Inform. Eng. (DaEng-2013)*, 2014, pp. 13–22.
[36] P. Machado, B. Fernandes, and P. Novais, "Benchmarking data augmentation techniques for tabular data," in *Int. Conf. Intell. Data Eng. Automat. Learn.*, Springer, 2022, pp. 104–112.
[37] M. Fallahian, M. Dorodchi, and K. Kreth, "GAN-based tabular data generator for constructing synopsis in approximate query processing: Challenges and solutions," 2022, *arXiv:2212.09015*.
[38] A. Buriro, F. Ricci, and B. Crispo, "SWIPEGAN: Swiping data augmentation using generative adversarial networks for smartphone user authentication," in *Proc. 3rd ACM Workshop Wireless Secur. Mach. Learn.*, 2021, pp. 85–90.

[39] Y. Elor and H. Averbuch-Elor, "To smote, or not to smote?" 2022, *arXiv:2201.08528*.
[40] J. Guan, X. Jiang, and B. Mao, "A method for class-imbalance learning in android malware detection," *Electronics*, vol. 10, no. 24, 2021, Art. no. 3124. doi: 10.3390/electronics10243124.
[41] H. Sahlaoui, E. A. A. Alaoui, S. Agoujil, and A. Nayyar, "An empirical assessment of smote variants techniques and interpretation methods in improving the accuracy and the interpretability of student performance models," *Educ. Inform. Technol.*, vol. 29, pp. 1–37, 2023.
[42] F. Specht, J. Otto, and D. Ratz, "Generation of synthetic data to improve security monitoring for cyber-physical production systems," in *2023 IEEE 21st Int. Conf. Indust. Inform. (INDIN)*, 2023, pp. 1–7.
[43] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019.
[44] O. Habibi, M. Chemmakha, and M. Lazaar, "Imbalanced tabular data modelization using ctgan and machine learning to improve iot botnet attacks detection," *Eng. Appl. Artif. Intell.*, vol. 118, no. 9, 2023, Art. no. 105669. doi: 10.1016/j.engappai.2022.105669.
[45] J. Li, J. He, W. Li, W. Fang, G. Yang and T. Li, "SynDroid: An adaptive enhanced Android malware classification method based on CTGAN-SVM," *Comput. Secur.*, vol. 137, 2024, Art. no. 103604. doi: 10.1016/j.cose.2023.103604.
[46] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inform. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009. doi: 10.1016/j.ipm.2009.03.002.
[47] F. T. Liu, K. M. Ting, and Z. -H. Zhou, "Isolation forest," in *2008 Eighth IEEE Int. Conf. Data Min.*, 2008, pp. 413–422.