Neural Computing and Applications (2024) 36:6955–6972
https://doi.org/10.1007/s00521-024-09439-x

ORIGINAL ARTICLE

Comparative study of ML models for IIoT intrusion detection: impact of data preprocessing and balancing

Abdulrahman Mahmoud Eid[1] · Bassel Soudan[1] · Ali Bou Nassif[1] · MohammadNoor Injadat[2]

Received: 10 October 2023 / Accepted: 15 January 2024 / Published online: 11 February 2024

Abstract

This study investigates the effectiveness of six prominent machine learning models—random forest, decision trees, K-nearest neighbor, logistic regression, support vector machines, and Naïve Bayes—for intrusion detection systems in industrial Internet of Things environments. The evaluation encompasses the effects of data preprocessing techniques, including feature engineering, data normalization, recoding, and missing data mitigation. Furthermore, the research delves into dataset balancing, examining the effects of six different techniques on model performance. The investigations are conducted using the domain-specific WUSTL-IIOT-2021 dataset, which captures the unique characteristics of IIoT data. The study also investigates multi-class attack identification utilizing an innovative SMOTE-based multi-class balancing approach to tackle dataset imbalances. The results indicate that data preprocessing and intelligent dataset balancing produce consistent enhancements in the classification performance of the selected models across binary and multi-classification tasks. Random forest emerges as the standout algorithm, delivering consistently high performance with computational efficiency.

Keywords Industrial Internet of Things · Intrusion detection · Machine learning algorithms · Data normalization · Dataset balancing

1 Introduction

Industrial Internet of Things (IIoT) networks play a crucial role in interconnecting diverse components of industrial control systems (ICS), enabling efficient monitoring of industrial processes and equipment. However, IIoT networks differ significantly from conventional IT networks [1, 2]. These disparities need to be thoroughly acknowledged when tackling the security challenges in IIoT networks. The unique behaviors and attributes of these two network types underscore the importance of tailored solutions that cater to the distinctive requirements of IIoT environments.

The proliferation of IIoT implementations across diverse industries has facilitated seamless communication between interconnected devices and systems [3, 4]. Nonetheless, this heightened connectivity has concurrently rendered IIoT networks more susceptible to cyberattacks [5–7]. In response, organizations have turned to the development of intrusion detection systems (IDSs) as a means to bolster the security of their IIoT networks [8–11]. These systems play a vital role in identifying potential cyber threats and enabling prompt responses to safeguard the integrity of IIoT networks.

1.1 Related work

In recent years, there has been growing interest in the literature in exploring the application of different ML algorithms for the development of IDS systems for mitigating attacks on IIoT networks [12]. Support vector machine model (SVM) was assessed for its potential in detecting

✉ Bassel Soudan
bsoudan@sharjah.ac.ae

Abdulrahman Mahmoud Eid
U20105546@sharjah.ac.ae

Ali Bou Nassif
anassif@sharjah.ac.ae

MohammadNoor Injadat
minjadat@zu.edu.jo

[1] Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah, UAE

[2] Department of Data Science and AI, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

compromises within an industrial system [13]. The researchers simulated a reconnaissance attack, where a command injection exploit was targeted at the ICS controller. The algorithm was implemented using MATLAB and validated on a simulated test bed employing the Tennessee Eastman (TE) dataset. While the researchers highlighted their model's real-time anomaly detection capabilities, the specific detection accuracy of their implementation was not explicitly reported.

Another study integrated the J48 and I Bayes approaches in the development of an IDS for IIoT networks [14]. The researchers used the labeled remote terminal unit (RTU) telemetry data from the gas pipeline system at Mississippi State University's Critical Infrastructure Protection Center (CIPC) as the dataset for training and evaluation. Experimental results achieved an accuracy of 99.5% for the specific-attack-labeled cases. However, no other performance metrics were investigated.

Vulfin et al. conducted a study focusing on detecting reconnaissance attacks on ICS systems [15]. They evaluated a number of ML algorithms such as random forest (RF), logistic regression (LR), and multilayer perceptron (MLP) using the WUSTL-IIOT-2018 dataset. Their proposed models achieved F1-scores ranging between 91 and 95% across different scenarios.

A study was undertaken to assess the effectiveness of diverse ML algorithms in detecting compromises triggered by command and injection attacks [16]. The researchers evaluated the I Bayes, RF, J48, non-nested generalized exemplars, SVM, and OneR algorithms. The models were trained and evaluated using the Mississippi State University's RTU dataset. The researchers reported that RF excelled in all classes, perfectly classifying normal responses, and achieving recall/precision values better than 75% for five response injection classes.

Researchers explored the use of SVM for identifying network traffic abnormalities [17]. Their investigation employed simulations based on the IEEE 118 bus network, effectively distinguishing between normal and fault scenarios. They reported that their IDS was able to maintain the system free from cyberattacks throughout the study. However, their highest-performing model could not achieve more than 95% accuracy. Unfortunately, the researchers did not consider any other performance metrics. Similarly, an anomaly-based IDS leveraging the SVM algorithm was assessed for intrusion detection in electric grid traffic, utilizing solely two attributes, "data rate" and "packet size" [18]. The training dataset was collected from a SCADA system during regular operation. The study reported a classification accuracy of 98.8%. However, it is essential to note that their training dataset lacked threat-related data, and the performance evaluation was based on accuracy exclusively.

Song et al. evaluated the ability of an IDS to detect availability compromises in an IIoT network [19]. The proposed IDS employed the extra trees classifier, RF, and K-nearest neighbor (KNN) algorithms. The models were trained using the WUSTL-IIOT-2018 dataset and validated using the cybersecurity Modbus ICS dataset. They reported that extra trees classifier and RF models achieved accuracies of 99.0%.

Zolanvari et al. in [20] evaluated multiple ML algorithms such as RF, decision trees (DT), KNN, LR, SVM, artificial neural networks (ANNs), and Naïve Bayes (NB) for the development of an IDS. Their models were trained and evaluated using the recently developed WUSTL-IIOT-2021 dataset [21]. The authors utilized several performance metrics to evaluate the efficacy of the proposed IDS. They reported that RF achieved the highest performance with an accuracy score of 99.99% and a Matthews correlation coefficient (MCC) score of 96.81%. Conversely, NB achieved 97.48% accuracy but only 24.4% MCC. They attributed the low performance of NB to the significant impact of dataset imbalance.

1.2 Gap analysis for existing works

Tables 1 and 2 present a comprehensive assessment of the literature related to the development of IDS systems for IIoT networks considering various aspects such as the ML models employed, and the datasets utilized for evaluation purposes. It can be seen that the collective literature suffers from the following gaps:

1. *Limited comparative analysis of the ML algorithms* Table 1 shows that each of the previous works concentrated on the evaluation of either one or a very limited number of models and that there is a deficiency in comparative analysis of the different algorithms. Such a comparative analysis would provide valuable

Table 1 Categorization of the IDS systems in the literature based on ML model

References	Machine learning model					
	SVM	LR	NB	RF	DT	KNN
[13]	X					
[14]					X	
[15]		X		X		
[16]	X		X	X		
[17]	X					
[18]	X					
[19]				X	X	X
[20]	X	X	X	X	X	X

Table 2 Categorization of the IDS systems in the literature based on dataset

References	Dataset used	Dataset access status	Comments
[13]	Tennessee Eastman dataset	Private	IIoT-specific
[14]	Telemetry data from the gas pipeline system	Public	IIoT-specific
[15]	WUSTL-IIOT-2018 dataset	Public	IIoT-specific
[16]	Telemetry data from the gas pipeline system	Public	Not IIoT-specific
[17]	IEEE 118 bus network simulations	Private	Not IIoT-specific
[18]	SCADA system during regular operation	Private	IIoT-specific
[19]	WUSTL-IIOT-2018 and cybersecurity Modbus ICS dataset	Public	IIoT-specific
[20]	WUSTL-IIOT-2021 dataset	Public	IIoT-specific

insight into the strengths and weaknesses of the individual algorithms and would allow identification of the most effective ML algorithms for developing robust and accurate IDS models.

2. *Limited dataset relevance* Table 2 shows that many of the examined works rely on outdated, non-domain-specific, private, or simulated datasets for training and evaluation purposes. While these models often demonstrate high accuracy within the limitations of their datasets, their efficacy in detecting novel cyberattack scenarios remains questionable. The lack of up-to-date and representative datasets poses a significant challenge in developing robust and reliable IDS models for IIoT systems.
3. *Absence of consideration for data preprocessing* it is noted that the majority of the reviewed works did not incorporate consideration for data preprocessing. It is well known that the efficacy of ML models is markedly influenced by the quality of input data furnished to the model [22, 23]. Hence, it is crucial to consider data preprocessing in the development of the IDS model to increase detection accuracy.
4. *Absence of consideration for multi-class identification of attacks* it is observed that the IDS systems proposed in the literature are mainly used for binary classification between normal and abnormal behavior. Very little work has been done on detection and classification of multiple attack scenarios.

In light of the aforementioned research gaps, this work endeavors to perform a comprehensive comparison between the six ML algorithms commonly used in the existing literature for the implementation of IDS systems for IIoT networks. This work will also investigate utilization of the recently developed WUSTL-IIOT-2021 dataset since it is more specific for the domain of IIoT networks. Furthermore, this work will evaluate the effect of data preparation strategies on the performance of the models from the literature. Finally, this work will investigate the performance of these techniques for multi-classification between different intrusion scenarios. In particular, this work extends beyond the work of reference [20], which used the default WUSTL-IIOT-2021 dataset without considering the effects of dataset balancing and also did not consider the possibility of discriminating the attack type through multi-class classification.

1.3 Contributions

The contributions of this work can be summarized into the following points:

1. Exploring the effectiveness of various dataset balancing techniques by assessing their impact on the performance of IDS models in detecting cyberattacks in IIoT networks.
2. Evaluating the performance of the leading six ML algorithms commonly used in intrusion detection research for IIoT using the domain-specific WUSTL-IIoT-2021. Dataset.
3. Evaluating the comparative performance of these six ML algorithms under varying conditions involving dataset normalization and balancing techniques.
4. Implementing a novel multi-class dataset balancing and investigating the effectiveness of the leading models from the literature in multi-classification of attacks on an IIoT network.

2 Background

This section will introduce some of the technical concepts that will be needed for the rest of the discussion. In particular, this section will give a brief introduction to the different ML algorithms used in the literature as well as an introduction to techniques used for balancing classes within a training dataset.

2.1 ML models commonly used for developing IDS systems for IIoT networks

The summary in Table 3 shows that researchers have opted to use mainly six ML algorithms for the development of IDS systems for IIoT networks [24]. A brief introduction and the main characteristics of these algorithm in relevance to intrusion detection in IIoT networks is summarized in Table 3.

2.2 Data balancing

Class imbalance is a common challenge affecting the performance of machine learning models where one class exhibits a higher attribute rate compared to others. This imbalance can significantly impact the accuracy of the model, particularly in classification operations. Optimal performance of most algorithms is observed when there is balanced representation across classes, aiming to minimize false rates and improve accuracy. However, when faced with imbalanced data, the model may excel in predicting the majority class but struggle with the minority class, which often holds greater significance. To address this issue, various algorithms have been developed for improving the class distribution within a dataset, as shown in Fig. 1. The following subsections will briefly introduce each of these techniques.

2.2.1 Cost-sensitive learning (CSL)

Cost-sensitive learning (CSL) as a means of inducing cost-sensitive trees, showcasing its practicality and ease of integration with existing ML algorithms [25]. The core principle involves assigning distinct weights to the minority classes, enabling the ML model to prioritize their importance. CSL is particularly useful in the context of IDS for IIoT due to its ability to address the inherent class imbalance that often characterizes intrusion detection datasets, thereby improving the accuracy of detecting critical cyber threats.

2.2.2 Data level preprocessing

Data level preprocessing involves manipulating the dataset to achieve a balanced distribution. As depicted in Fig. 1, this can be achieved by either oversampling the minority classes or under-sampling the majority classes.

2.2.2.1 Under-sampling Under-sampling involves randomly removing instances from the majority class until a balanced distribution is achieved. While under-sampling effectively improves class distribution and helps address the challenges posed by skewed datasets, it may lead to information loss due to the removal of data points. Careful consideration of the trade-offs between improved balance and potential loss of data patterns is essential when applying under-sampling techniques to enhance the performance of IDS models in IIoT networks.

Table 3 The highlights of the chosen ML models for IIoT IDS

ML model	Description
Decision tree (DT)	A classification technique converting root-to-leaf paths into rules, suitable for reducing data uncertainty. Noteworthy for transparent decision-making and adept at detecting anomalies, it finds utility in analyzing intricate data patterns for IIoT IDS systems
Random forest (RF)	An ensemble method employing multiple decision trees, mitigating overfitting. Its efficiency on extensive datasets makes it valuable for handling the intricacies of IIoT network traffic and intricate relationships required for effective intrusion detection
Support vector machine (SVM)	SVMs classify through hyperplanes, excelling in linear classification. Their adeptness at handling high-dimensional data and intricate decision boundaries enhances intrusion detection accuracy in IIoT contexts
K-nearest neighbors (KNN)	KNN, a non-model-based approach, classifies based on proximity to neighboring data points. Its ability to capture subtle deviations from normal network behavior proves invaluable for anomaly and intrusion detection in IIoT scenarios
Logistic regression (LR)	Logistic regression models event probabilities from input features, suitable for anomaly and intrusion detection. Its adaptability for multi-class tasks, simplicity, and capacity to handle large datasets make it useful for IIoT IDS implementation
Naïve Bayes (NB)	Naïve Bayes is an efficient algorithm for categorical data, aligning well with IIoT's network traffic attributes. Despite assuming feature independence, it excels in quick and reliable anomaly detection by calculating probabilities based on attribute occurrences

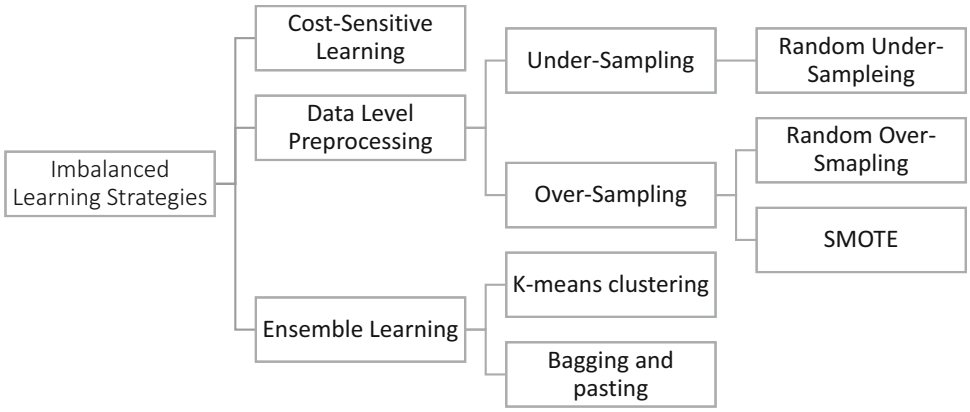

Fig. 1 Techniques for handling imbalanced data

2.2.2.2 Over-sampling On the other hand, over-sampling operates by increasing the representation of the minority class. This can be achieved through attribute duplication or synthetic minority oversampling technique (SMOTE).

- In attribute duplication, attributes from the minority class are replicated to match the number of instances in the majority class. This essentially involves creating copies of existing data points from the minority class. While this method may help balance class distribution, it carries the risk of introducing redundancy and potentially overfitting the model to the minority class, which can lead to reduced generalization capabilities.
- SMOTE on the other hand addresses imbalanced class distribution by generating synthetic attributes for the minority class [46]. Synthetic samples are inserted along line segments connecting each minority class instance with its K-nearest neighbors [25]. The newly created instances contribute to a more balanced class distribution while also introducing diversity into the dataset. SMOTE can help alleviate the risk of overfitting and improve the ability of the model to recognize complex decision boundaries, thus enhancing its generalization performance.

Both over-sampling techniques can be advantageous in addressing class imbalance, but as with under-sampling, careful consideration is needed to avoid potential drawbacks. Over-sampling may increase the risk of overfitting, and synthetic instances generated by SMOTE should reflect meaningful patterns present in the data to ensure the model's improved performance on unseen samples.

2.2.3 Ensemble learning

2.2.3.1 K-means clustering Cluster-based majority under-sampling was developed for choosing a subset from the majority class that is representative [26]. As compared to random under-sampling, cluster-based under-sampling can efficiently prevent loss of crucial majority class information. This technique uses cluster centroids as reduced data points. Therefore, instead of directly sampling from the data, it generates artificial samples that represent the majority class.

2.2.3.2 Bagging and pasting In [27], researchers introduced the concept of bootstrap aggregating to construct ensembles, whereby multiple classifiers are trained using bootstrapped copies of the original training dataset. This involves generating new datasets by randomly selecting instances from the initial dataset.

3 Methodology

The diagram in Fig. 2 demonstrates the main step of the implementation and evaluation methodology for this work. First, multiple data cleaning steps were applied to the dataset to enhance its quality and reliability. Subsequently, categorical features were encoded using one-hot encoding enabling their conversion into a suitable representation for subsequent analysis. Simultaneously, the data types of numerical features were standardized to achieve uniformity across the dataset. Following these transformations, both categorical and numerical features underwent a normalization process, thereby facilitating consistent and comparable scaling of the data.

Upon completion of the data preprocessing stage, the dataset was partitioned into an 80% training set and a 20% testing set. To address class imbalance and mitigate bias, six balancing algorithms were applied to the training set, while keeping the testing set intact. Subsequently, six ML models were constructed using the training set, and their performance was evaluated using the testing set. To assess the effectiveness of the proposed model, various evaluation metrics were calculated based on the prediction results of the various ML models. The following subsections will discuss the steps of the methodology in detail. But, first a discussion of the selected dataset and its characteristics.

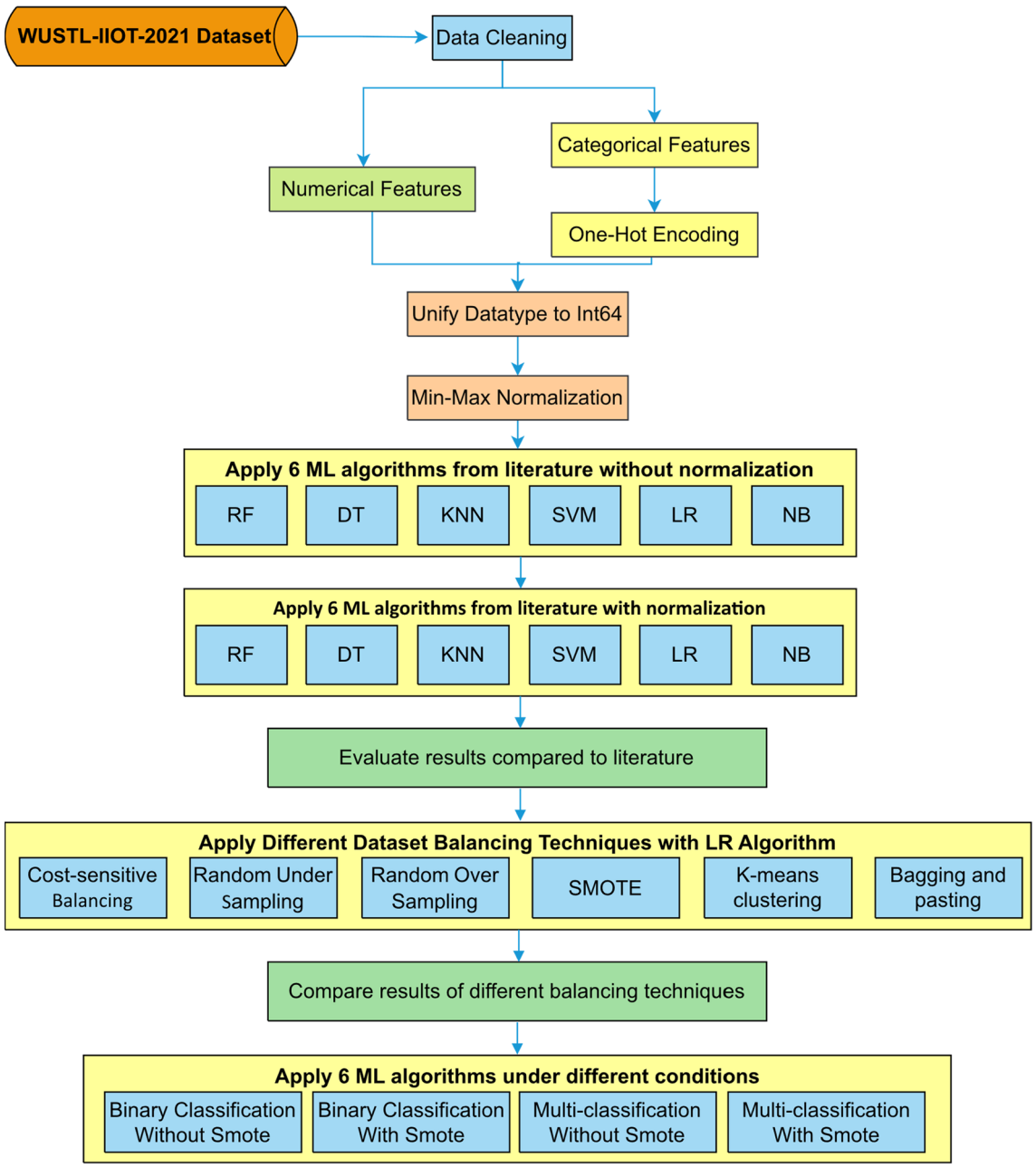

Fig. 2 Flowchart for the evaluation methodology

3.1 Dataset

The selection of the WUSTL-IIoT-2021 dataset for this study is underpinned by several compelling justifications. Firstly, this dataset is specifically curated for IIoT networks. It was designed to mimic real-world industrial operations encompassing a wide range of network activities encountered in industrial environments [20]. This contextual relevance ensures that the dataset captures the challenges, intricacies, and real-world nuances of IIoT environments.

Moreover, the WUSTL-IIoT-2021 dataset consists of normal and attack traffic, covering four distinct attack scenarios: DoS, reconnaissance, command injection, and backdoor attacks. This diversity facilitates a holistic evaluation of the performance across a spectrum of scenarios. Furthermore, the availability of ground truth labels for the dataset provides a solid foundation for evaluating and benchmarking the performance. The presence of labeled instances allows quantifying metrics like accuracy, precision, recall, and F1-score.

Table 4 illustrates the distribution of the dataset across the different classes, while Fig. 3 presents the binary distribution of the dataset between normal and attack instances. All records from the various attacks were grouped under "Malicious."

3.2 Data preprocessing and cleaning

Effective data preprocessing and cleaning are pivotal in ML-based modeling. As the quality of the produced model relies heavily on the quality of the training data. To ensure optimal dataset preparation, it is essential to go through a number of data cleaning and transformation processes [28].

3.2.1 Data cleaning

Data cleaning involves organizing, editing, and refining data to ensure consistency and analysis readiness. It eliminates inaccurate or irrelevant data, enabling effective interpretation and analysis [29]. A three-step approach was implemented for data cleaning:

3.2.1.1 Remove irrelevant data and feature selection To ensure better model generalization and eliminate irrelevant data, specific features including "StartTime," "LastTime," "SrcAddr," "DstAddr," "sIpId," and "dIpId" were removed from the dataset. These features uniquely identified the attacks, potentially compromising the model's ability to handle unseen data. Additionally, non-essential elements such as hashtags, URLs, emoticons, HTML tags, and dates were eliminated during the data cleaning process [20].

3.2.1.2 Remove duplicated records Eliminating duplicates is essential to achieve balanced outcomes, as they can increase storage needs, hinder analysis, and bias the results of the ML models.

Table 4 Distribution of traffic in the WUSTL-IIoT-2021 dataset

Traffic type	Number of records	Percentage (%)
Normal traffic	972,137	92.59
DoS	68,811	6.56
Reconnaissance	7220	0.69
Command injection	228	0.02
Backdoor	179	0.02
Total records	1,048,575	100

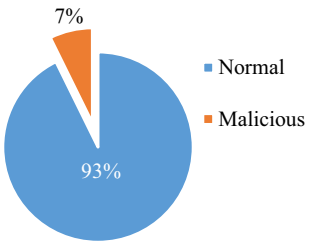

Fig. 3 Binary distribution of the WUSTL-IIOT-2021 dataset (normal vs. malicious)

3.2.1.3 Handling missing data Given the abundance of records available in the dataset, all records with missing data were dropped from the dataset.

3.2.2 Data transformation

Data transformation is performed to optimize outcomes. In IIoT intrusion detection, data transformation involves two main steps.

3.2.2.1 Data type conversion The dataset contained two main types of features: categorical and numerical. It became necessary to modify the representation of these features to establish consistent representation and facilitate efficient computation and analysis. The following modifications were made to the representation of the different types of features:

a. Categorical features were encoded using one-hot encoding to convert them into a numerical format suitable for normalization and model training. The process involves representing each unique category as a binary vector, where only one element corresponding to the category is marked as a 1, while all other elements are set to 0. This transformation ensures that the categorical information is maintained and does not introduce ordinal relationships between categories. After this encoding step, categorical features can be treated in a similar manner to numerical features.
b. Numerical values (numerical features and the encoded categorical features) were all unified to an "Int64" variable size, ensuring uniformity and consistency in the representation of the data.

3.2.2.2 Data normalization Normalization is the process of rescaling numerical attributes to a common range, typically centered between 0 and 1. In this study, we employed Min–Max normalization, which offers several advantages, to normalize all attributes. This normalization technique ensures that the features are scaled proportionally and preserves the relationships between the data points, contributing to improved accuracy.

3.3 Training and testing procedure

To ensure randomness, a shuffling process was applied to the dataset before it was split into the separate training and testing datasets. The training dataset comprised 80% of the original data, while the testing dataset comprised the remaining 20%. This partitioning strategy allowed for an unbiased evaluation of the model's performance on unseen data, enabling the assessment of its ability to generalize and make accurate predictions.

3.4 Evaluation metrics

The performance evaluation of the six proposed models in this study encompassed a comprehensive set of five evaluation metrics: accuracy, recall, precision, *F*1-score, and MCC. These metrics were selected to provide a thorough assessment of each model's effectiveness and the performance of the underlying ML algorithms employed. Moreover, the evaluation process included the measurement of training and testing times for the multi-classification IDS. By considering these different figures of merit, a detailed and precise evaluation of each model was obtained, facilitating a comprehensive comparison of their performance. This evaluation framework enables the identification of the most suitable ML algorithm for a given model, based on the specific objectives and priorities established for the system.

One of the most significant indicators of performance for ML models is the confusion matrix which reflects the model's achievement in correctly classifying records. The details in Table 5 show the definition of the terms for the confusion table that are used in our work.

The goal for our IDS is to reduce the false negative (FN) and increase the true-negative (TN) classifications. The definition for these terms is described as follows:

- True negatives (TNs): the number of legitimate packets that have been classified successfully as normal.
- True positives (TPs): the number of abnormal or malicious packets that have been successfully classified as an attack.
- False positive (FP): the number of non-attack packets that have been wrongly classified as an attack.
- False negative (FN): the number of anomalous packets that were wrongly classified as legitimate packets.

Additionally, we used the following figures of merit to evaluate the performance of the different ML models:

Accuracy: accurately predicted samples as a proportion of total predictions, it is calculated based on the expression in Eq. (1):

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

Recall: the percentage of malicious traffic that has been correctly classified as an attack. It is also known as sensitivity or true positive rate (TPR). It is calculated based on the expression in Eq. (2):

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2}$$

Precision: the percentage of relevant results. It is mainly used when the FP is a priority, and it can be calculated based on the expression in Eq. (3):

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{3}$$

*F*1-score: the harmonic mean between precision and recall. It is used when both recall and precision are a priority. It is calculated based on the expression in Eq. (4):

$$\mathrm{F1-Score} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Matthews correlation coefficient (MCC): this statistical metric gauges the quality of classification by illustrating the agreement between observed and expected values. The MCC is a robust measure that yields a high score when the predictions consistently exhibit strong performance across all facets of the confusion matrix [30, 31]. It will be utilized to comprehensively assess the model's performance in binary classification. It is calculated based on the expression in Eq. (5):

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FN)}} \tag{5}$$

While a high accuracy rate may be expected in an imbalanced dataset where most samples are legitimate traffic, it does not necessarily indicate a strong model. Thus, accuracy alone is not a reliable measure in the presence of dataset imbalances. MCC on the other hand is a reliable statistical measure that accounts for the model's performance across all areas of the confusion matrix, providing a more comprehensive evaluation of effectiveness.

Table 5 Definition of the confusion matrix parameters

	Legitimate traffic (0)	Malicious traffic (1)
Predicted negative (legitimate traffic) (0)	True negative (TN)	False positive (FP)
Predicted positive (malicious traffic) (1)	False negative (FN)	True positive (TP)

4 Results and discussion

The aim of this work is to investigate the effect of data normalization and dataset balancing on the performance of the leading ML models used for detecting malicious intrusions in IIoT networks in the literature. The diagram in Fig. 2 showed the operational flow of the evaluations conducted in this work, and the following subsections will present a discussion of the results.

4.1 Effect of data normalization

The first experiment conducted was to evaluate the effect of data normalization on the performance of the six ML models used in the literature. The models were implemented, and their performance was evaluated using the default un-normalized dataset, and the dataset after normalization as described in Sect. 3.2.2. The chart in Fig. 4 shows the MCC results for the different models as reported in reference work [20]. These results will be used as the benchmark since they represent the latest IDS built using the WUSTL-IIOT-2021 dataset.

4.1.1 Comparative performance of our implementation on the un-normalized dataset

The dataset was first taken through the cleaning steps described in Sect. 3.2.1 before it was partitioned into the training and testing datasets as described in Sect. 3.3. Then, the six ML models were implemented using Python, and trained on the un-normalized dataset. The results in Fig. 5 provide a comparison between the results of our implementation and the results from the benchmark. It is to be noted that the results of our implementation are based on the un-normalized dataset, while the results from the benchmark are based on a normalized dataset. This should explain the reduction in performance for SVM and LR in our implementation since they exhibit suboptimal performance on un-normalized datasets. It is observed that RF, DT, and KNN achieved a high classification accuracy (with MCC levels around 99%), while SVM, LR, and NB did not perform as well. Notably, our implementation yielded higher classification accuracy for RF, DT, KNN, and NB compared to the benchmark implementation. The performance of SVM, LR, and NB will be closely monitored to assess potential improvements and enhancements in their performance. To provide a comprehensive evaluation, all five performance metrics were calculated and are presented in Table 6. The findings demonstrate the performance consistency of the six ML algorithms across different evaluation metrics, ensuring a comprehensive assessment of their effectiveness in intrusion detection.

The superior performance of our implementation in comparison to the reference can be attributed to meticulous data preprocessing steps, including feature reduction, duplicate removal, and handling of missing data. Tailored data transformations were applied, such as one-hot encoding for categorical data and direct transformation of numerical features into integers. These measures optimized the dataset for the implementation and evaluation of the six selected ML algorithms.

4.1.2 Effect of normalization

"MinMaxScaler" technique was applied to the dataset, then the same six ML algorithms were employed to construct multiple IDS models. Figure 6 presents a comparison of the MCC results obtained from the benchmark, our implementation without dataset normalization, and our implementation after dataset normalization. The results highlight notable improvements in the performance of SVM and LR, both reaching an accuracy level better than 99% after normalization. However, the performance of the NB algorithm only exhibited a marginal increase compared to the un-normalized dataset, suggesting its limited ability to handle imbalanced datasets effectively. Table 7 presents the five performance metrics for our implementations using the normalized dataset.

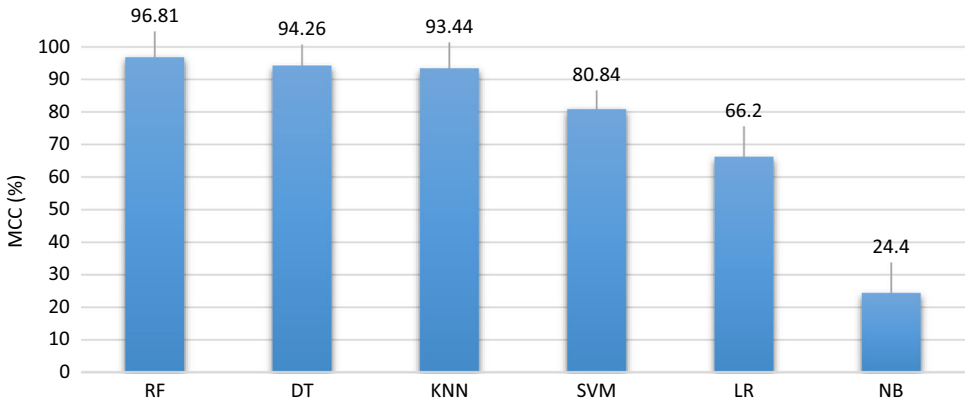

Fig. 4 MCC results of the six ML models from the literature [20]

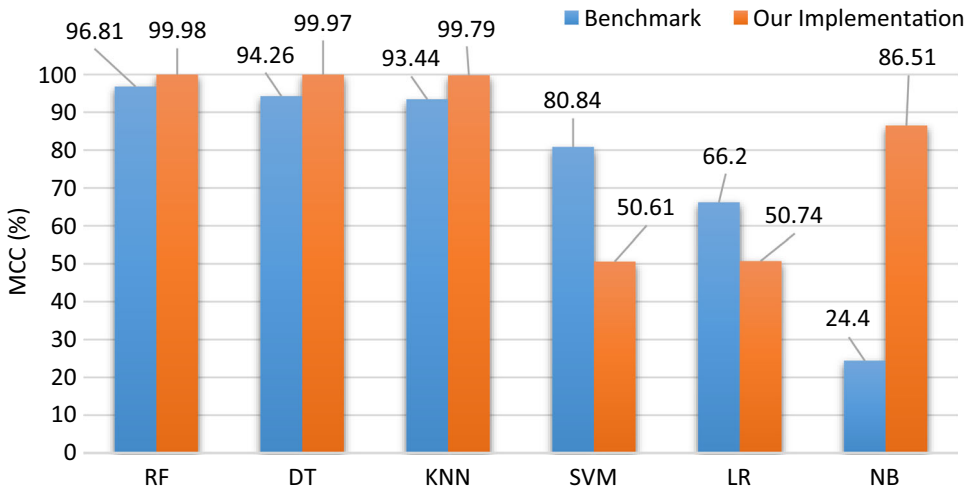

Fig. 5 MCC results of the six ML models from the benchmark and our implementation using the WUSTL-IIOT-2021 dataset without normalization

Table 6 Performance of ML algorithms on the un-normalized WUSTL-IIOT-2021 dataset

Performance metric	RF (%)	DT (%)	KNN (%)	SVM (%)	LR (%)	NB (%)
Accuracy	99.99	99.99	99.97	95.10	94.62	98.08
Precision	99.99	99.98	99.90	90.24%	89.34	91.78
Recall	99.98	99.98	99.89	69.91	66.36	94.77
*F*1-score	99.99	99.98	99.89	76.09	72.37	93.21
MCC	99.98	99.97	99.79	56.61	50.74	86.51

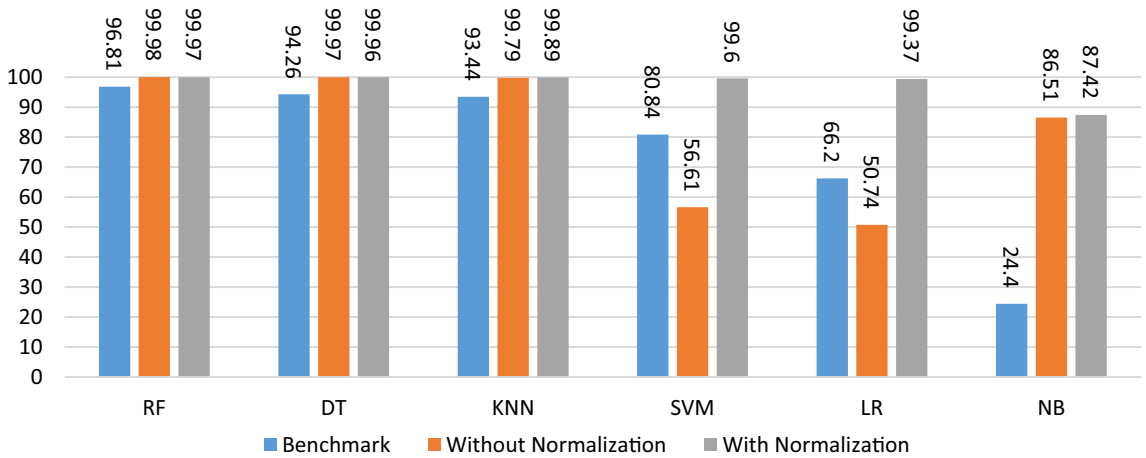

Fig. 6 MCC results of the six ML models from the benchmark and our implementation with and without normalization

Table 7 Performance of ML algorithms on the normalized WUSTL-IIOT-2021 dataset

Performance metric	RF (%)	DT (%)	KNN (%)	SVM (%)	LR (%)	NB (%)
Accuracy	99.99	99.99	93.98	99.95	99.91	97.95
Precision	99.99	99.98	99.95	99.89	99.82	89.09
Recall	99.97	99.99	99.95	99.72	99.56	98.87
*F*1-score	99.99	99.98	99.80	99.80	99.69	93.31
MCC	99.97	99.97	99.60	99.60	99.37	87.42

The results show that the performance of all ML models improved after the application of data normalization. Particularly, the performance of SVM and LR improved significantly across all metrics. Accordingly, it was decided

to utilize the MinMaxScaler technique for dataset normalization in all further experiments and model development.

4.2 Effect of different balancing techniques on the performance of LR algorithm

The severe class imbalance observed in the WUSTL-IIOT-2021 dataset, as highlighted in Table 4, poses a challenge to the learning process and can lead to biased machine learning models. To address this issue, we conducted a comprehensive investigation into the effects of various balancing techniques on the performance of the IDS. Among the ML models, LR was selected for its simplicity and interpretability, which makes it a suitable candidate for initial investigations into the dataset's class imbalance and the effects of various balancing techniques. The evaluated balancing techniques include cost-sensitive learning (CSL), random under-sampling, random over-sampling, SMOTE, and bagging and pasting. Unfortunately, the K-means clustering technique could not be evaluated due to the dataset's large size, rendering it impractical.

It is to be noted that the primary objective of an IDS is to accurately detect security breaches in real-time, with minimal overlooked threats. Therefore, it is paramount to keep false negatives (undetected malicious traffic) at a minimum. Consequently, the recall and MCC performance metrics become critical determinants in identifying the most effective technique. Therefore, these metrics were used to evaluate the relative performance of the models with the application of the different balancing techniques. The results presented in Fig. 7 show that all balancing techniques produced an improvement in the recall of the model. Additionally, SMOTE seems to have an advantage over the other techniques as summarized in the following points:

1. Effective MCC: SMOTE achieves the highest MCC score of 99.37%. This indicates that SMOTE is highly successful in detecting malicious traffic and minimizing false negatives.
2. Resource efficiency: unlike random under-sampling and bagging and pasting, SMOTE does not sacrifice data samples or impose excessive computational costs. It generates artificial data points that are marginally different from the original data, allowing for efficient utilization of available resources.
3. Scalability and accuracy: SMOTE's ability to generate synthetic data without introducing duplicates makes it an ideal balancing algorithm for classification tasks that prioritize accuracy and scalability. It strikes a balance between maintaining data diversity and avoiding overfitting, resulting in improved overall model performance.

These findings highlight the effectiveness of SMOTE in achieving a balance between accurate classification, efficient resource utilization, and scalability compared to other balancing techniques. Therefore, it was decided to use SMOTE in all further experiments.

4.3 Effect of SMOTE on the performance of the ML models from the literature

An experiment was conducted to assess the impact of SMOTE balancing on the performance of the six ML models from the literature in binary intrusion detection (normal versus attack). The WUSTL-IIoT-2021 dataset, which has already been normalized, was split into a training dataset (randomly selected 80% of the samples)

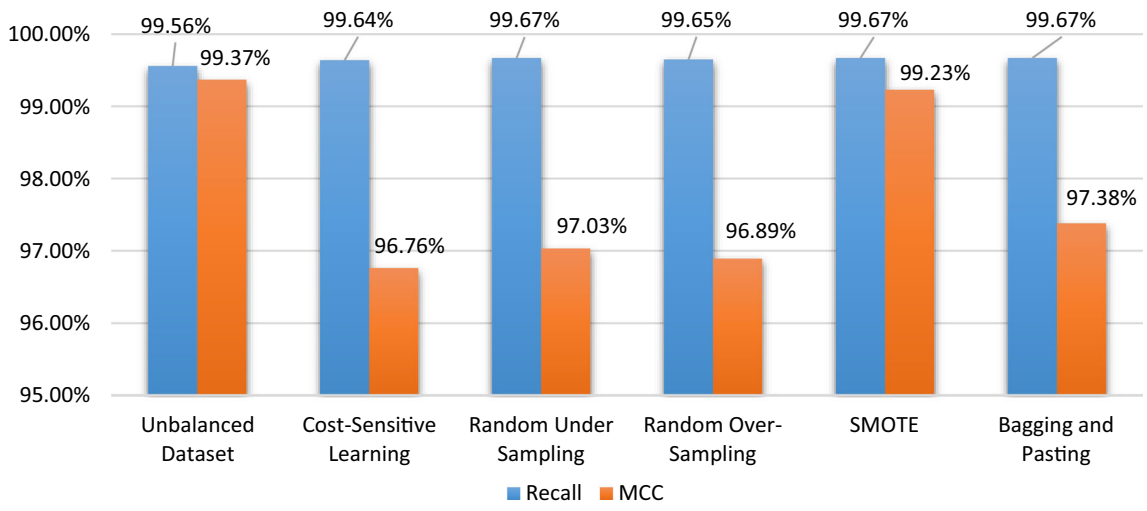

Fig. 7 Comparison of balancing techniques based on recall and MCC score

and a testing dataset (remaining 20% of the samples). The training dataset (comprising of 838,903 samples) underwent SMOTE balancing to achieve a balance between normal and malicious samples (as outlined by Table 8). On the other hand, the testing dataset (comprising of the remaining 209,718 samples) was left unbalanced to represent the expected traffic in an IIoT network under attack.

Figure 8 illustrates the percentage distribution of the dataset among the various classes following the application of the SMOTE balancing technique. Since the first model involves binary classification, all classes were consolidated into two categories: "Normal" and "Attack," with an equal number of attributes for each class.

All six ML models were trained using the balanced training dataset and subsequently evaluated using the unbalanced testing dataset. The charts in Fig. 9 present the impact of SMOTE balancing on the evaluation metrics of the different algorithms. The charts demonstrate that most algorithms either maintained or exhibited improved performance with the application of SMOTE.

Notably, the NB algorithm demonstrated significant improvement across all metrics as a result of SMOTE balancing. This enhancement can be attributed to the algorithm's reliance on the assumption of feature independence when calculating class probabilities. In cases where the dataset is imbalanced, NB may exhibit a bias toward the majority class. However, by applying SMOTE to the dataset, synthetic samples are generated specifically for the minority class, which enables NB to capture a more comprehensive representation of the minority class, leading to improved accuracy in classifying instances from this class.

On the other hand, it is worth noting that the LR algorithm displayed a slight decrease in performance across various metrics when SMOTE balancing was applied. This can be attributed to the potential introduction of noise or outliers through the synthetic samples, which may disrupt the LR model's assumed linear relationships.

Table 9 summarizes the improvement in the MCC score for the different ML algorithms after SMOTE was applied to the dataset. The results demonstrate the remarkable improvement in the performance of the NB and LR algorithms.

4.4 Performance of the six ML algorithms in multi-classification of intrusion attacks

The preceding experiments showcased that the proposed combination of normalizing and balancing the dataset enhances the abilities of the models in determining whether the network is under attack or behaving normally. However, once it is established that an attack is underway, it becomes imperative to identify the type of attack so that the operators can tailor their response appropriately. Accordingly, an experiment was conducted to evaluate the performance of the ML models in a multi-class categorization of the type of attack to which the network is being subjected. To the best of our knowledge, no existing work has addressed the challenge of balancing multi-attack IIoT datasets or developed a multi-classification IDS specifically designed for IIoT environments.

The six ML algorithms from the literature were used to develop multi-classification IDS models for identifying the various attacks present in the WUSTL-IIoT-2021 dataset. The dataset underwent the same data preprocessing, cleaning, feature selection, and normalization procedures applied in the earlier binary classification experiments. In the absence of comparable benchmarks within existing research, the performance of these models will be assessed against each other to establish their relative effectiveness. The following subsections will present the results using the unbalanced versus SMOTE-balanced datasets.

Table 8 Distribution of samples in the training dataset balanced for binary classification using SMOTE

Category	Class distribution in default dataset		Selected samples for training dataset (80%)		SMOTE-balanced training dataset	
	Count	Percent	Count	Percent	Count	Percent
Normal traffic	972,137	92.59	777,752	92.71	777,752	50.00
DoS	68,811	6.56	55,000	6.56	194,440	12.50
Reconnaissance	7220	0.69	5773	0.69	194,440	12.50
Command injection	228	0.02	184	0.02	194,440	12.50
Backdoor	179	0.02	194	0.02	194,440	12.50
Total samples	1,048,575	100	838,903		1,555,512	

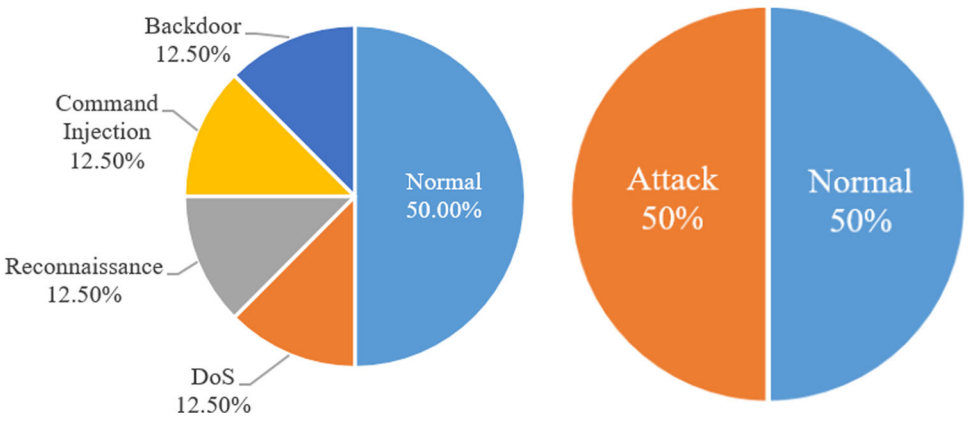

Fig. 8 Distribution of the training dataset after applying SMOTE for binary classification

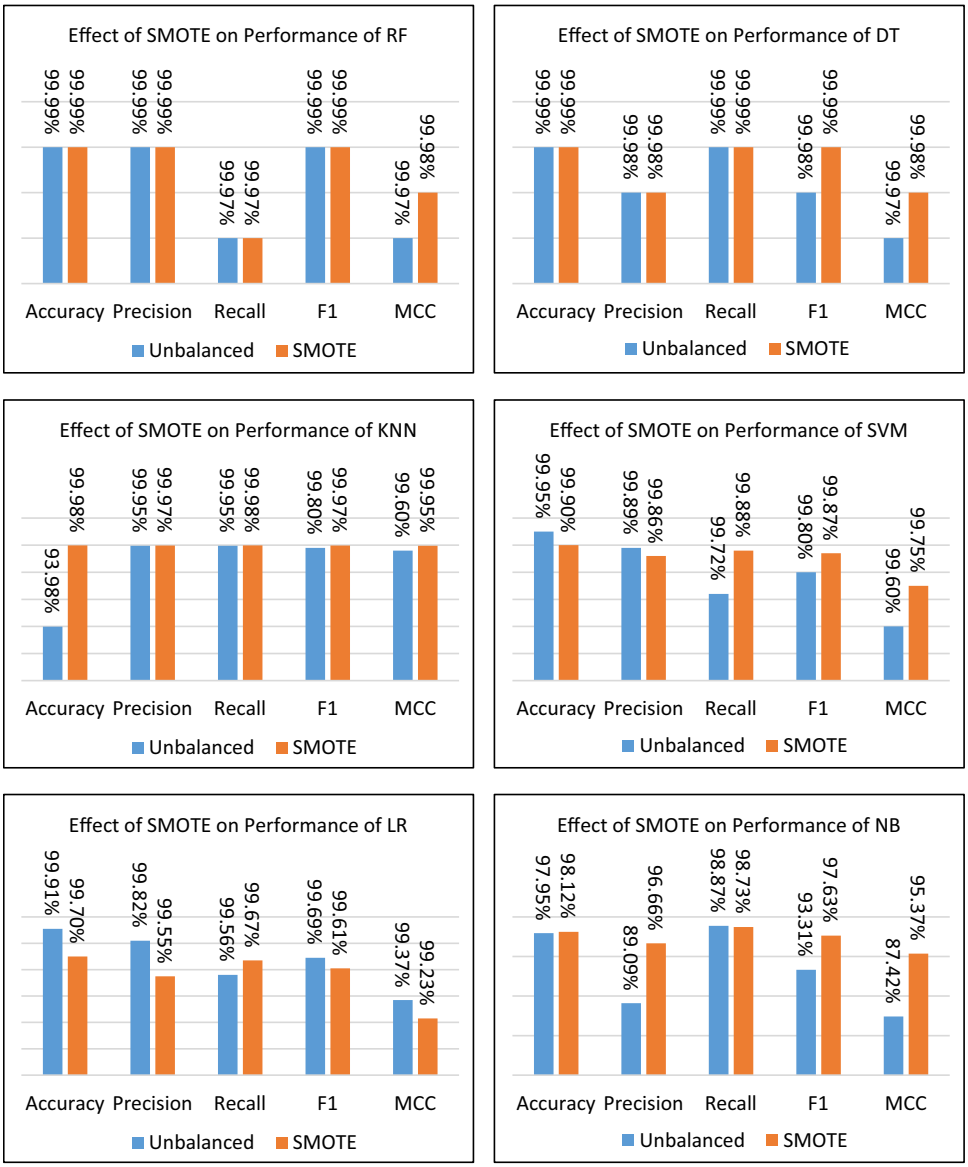

Fig. 9 Effect of SMOTE balancing on the performance of binary classification using ML models from the literature

Table 9 MCC results for binary classification using the different ML algorithms

Model	RF (%)	DT (%)	KNN (%)	SVM (%)	LR (%)	NB (%)
Benchmark (reference [20])	96.81	94.26	93.44	80.84	66.20	24.40
Without normalization	99.98	99.97	99.79	56.61	50.74	86.51
Normalization	99.97	99.96	99.89	99.6	99.37	87.42
With SMOTE	99.98	99.98	99.95	99.75	99.23	95.37

4.4.1 Performance of the ML algorithms in multi-classification using the unbalanced dataset

A portion of the default unbalanced WUSTL-IIoT-2021 dataset was extracted to create the training dataset for the multi-classification models, and their performance was evaluated using the performance metrics discussed in Sect. 3.4. Additionally, training and testing times were documented as an additional figure of merit for the different models.

In general, the results in Table 10 show that the ML algorithms produced reasonably consistent performance across the various evaluation metrics indicating reasonable effectiveness in detecting the different attack scenarios present in the dataset. Notably, RF and DT exhibited good resilience even though the dataset is severely unbalanced, while the performance of KNN, SVM, and LR is slightly lagging indicating possible room for improvement through dataset balancing. As anticipated from binary classification, NB exhibited relatively lower performance compared to the other models. A comparison of the processing time requirement seems to indicate that DT requires the least training and testing times among the models that produced good classification performance.

4.4.2 Effect of SMOTE balancing on performance of ML algorithms for multi-classification

The WUSTL-IIoT-2021 dataset comprises of five classes, four types of cyberattacks along with normal traffic. Accordingly, SOMTE was used to balance the training dataset to ensure equal representation of the classes. The minority classes (attack samples) underwent SMOTE-based up-sampling until their representation in the training dataset became the same as the majority class (normal traffic), as shown in Table 11.

The results presented in Table 12 show the multi-classification performance of the six ML algorithms with the use of the SMOTE-balanced training dataset using the different measures.

Figure 10 shows a comparison between the multi-classification performance of the different models before and after applying SMOTE balancing. The figure shows that RF produced identically high performance in both the unbalanced and balanced scenarios, indicating its capability to handle class imbalance effectively. Furthermore, Table 12 shows that it exhibited the same execution efficiency, as there was no significant increase in either training or testing times. Similarly, DT demonstrated the same outstanding performance in both scenarios. However, it is important to note that DT exhibited an increased training and testing time with the balanced dataset, which can be attributed to the significantly larger training dataset.

KNN on the other hand showed significant performance improvement after the application of balancing. However, this improvement comes at the cost of significant increase in testing time. Similarly, SVM showed impressive performance improvement, but at the expense of multiple orders of magnitude increase in both training and testing time. LR also witnessed improvements in all evaluation metrics with a relatively small increase in training time (from one minute to 3.5 min), and testing time (from one second to five seconds). Lastly, NB, which initially showed limitations in handling imbalanced datasets, demonstrated noticeable improvements in all metrics after the dataset was balanced, but with slight increases in both training and testing time.

Table 10 Multi-classification performance of ML algorithms using unbalanced dataset

Performance metric	RF	DT	KNN	SVM	LR	NB
Accuracy	99.99%	99.99%	99.97%	99.94%	99.85%	97.89%
Precision	98.96%	98.86%	96.90%	99.92%	93.60%	65.26%
Recall	98.15%	99.14%	91.15%	93.06%	86.47%	96.10%
*F*1-score	98.55%	98.98%	93.78%	96.17%	89.50%	72.68%
MCC	99.97%	99.96%	99.83%	99.61%	98.95%	87.21%
Training time (s)	127	10	5	643	60	10
Testing time (s)	3	2	1225	67	1	1

Table 11 Distribution of samples in the training dataset balanced for multi-classification using SMOTE

Category	Class distribution in default dataset		Selected samples for training dataset		SMOTE-balanced training dataset	
	Count	Percent	Count	Percent	Count	Percent
Normal traffic	972,137	92.59	777,752	92.71	777,752	20.00
DoS	68,811	6.56	55,000	6.56	777,752	20.00
Reconnaissance	7220	0.69	5773	0.69	777,752	20.00
Command injection	228	0.02	184	0.02	777,752	20.00
Backdoor	179	0.02	194	0.02	777,752	20.00
Total samples	1,048,575		838,903		3,888,760	

Table 12 Multi-classification performance of ML algorithms using SMOTE-balanced dataset

Performance metric	RF	DT	KNN	SVM	LR	NB
Accuracy	99.99%	99.99%	99.99%	99.94%	99.97%	95.56%
Precision	98.96%	98.86%	99.99%	99.94%	98.94%	95.98%
Recall	98.15%	99.14%	99.99%	99.94%	98.95%	95.43%
*F*1-score	98.55%	98.98%	99.99%	99.94%	98.94%	95.33%
MCC	99.97%	99.96%	99.99%	99.92%	98.72%	94.61%
Training time (s)	127	122	38	112,100	203	31
Testing time (s)	3	7	17,510	3320	5	9

In general, it can be concluded from these results that, with the application of dataset normalization and balancing using SMOTE, all six of the ML models from the literature produce reasonable performance in detecting and correctly classifying the different attacks present in the domain-specific WUSTL-IIoT-2021 dataset. Overall, KNN and SVM produced the best results based on the different figures of merit. However, at severe training and testing time penalties.

On the other hand, DT and LR produced highly competitive performance, approaching 99% in all metrics, albeit at the expense of a tenfold increase in training and testing durations. Notably, RF stands clearly as the superior algorithm amongst these six, consistently delivering robust performance even in the face of the significantly expanded dataset, while maintaining computational efficiency unaffected by the dataset's expansion. Applying optimizations to some of these techniques would further improve their performance [32].

5 Conclusion

This work presents a comprehensive quantitative analysis of the performance of RF, DT, KNN, LR, SVM, and NB in detecting intrusions in IIoT networks. These models are chosen due to their extensive use in relevant literature.

This work also investigated the effect of data preprocessing, namely feature engineering, data normalization, recoding, and mitigation of missing data on the performance of these models. The results reveal a substantial improvement in intrusion detection accuracy across all models following the application of the data preprocessing steps. This improvement ranged from 3% for the RF model to 260% for NB compared to the baseline from the literature, based on the MCC evaluation metric.

In addition, the work performed extensive analysis on the effect of six different dataset balancing techniques on the performance of the models, especially with the use of the domain-specific WUSTL-IIOT-2021 dataset. The results showed that balancing the dataset using SMOTE led to the highest binary classification effectiveness between normal and attack traffic. The effectiveness of the different models increased to the range of 95.4% (for NB) to 99.98% (for RD and DT).

Furthermore, this work implemented multi-class identification between the different attack types present in the dataset. A novel implementation of multi-class balancing using the SMOTE technique was used to counter the substantial imbalance in the dataset against some attack types. The results showed that RF, DT, and LR produced superior multi-class attack identification performance approaching 99.99% accuracy, while maintaining a very robust computation time. The other models produced highly competitive performance, however at the expense of a significant

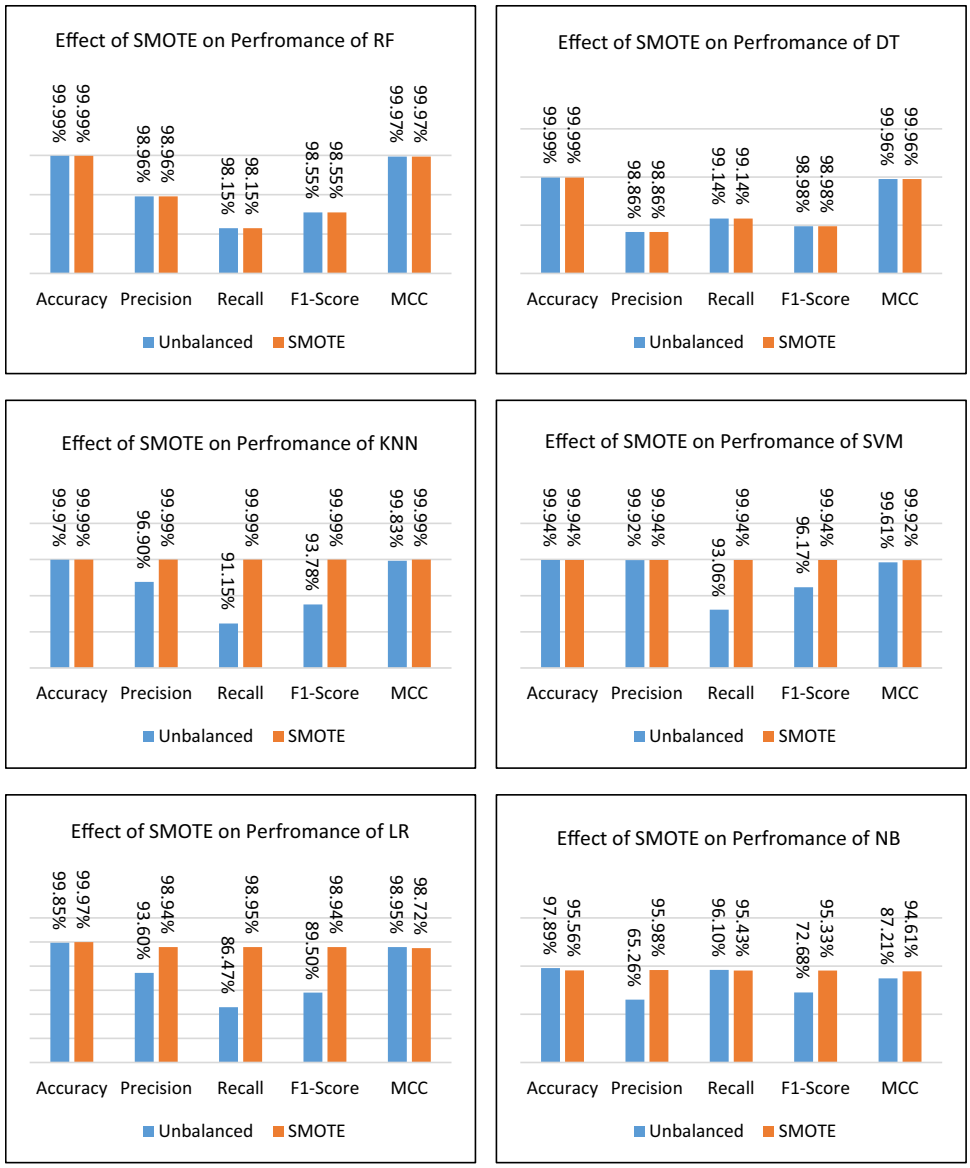

Fig. 10 Effect of SMOTE balancing on the performance of multi-classification using ML models from the literature

increase in training and testing times due to the considerably larger training dataset.

In general, it can be concluded that there was a performance improvement for all six ML models from the literature with the application of dataset normalization and balancing using SMOTE. However, for some models, this improvement came at considerable training and testing time penalties. Notably, RF stands clearly as the superior algorithm as it produces consistently high performance for both binary- and multi-classification while maintaining robust computational efficiency.

In conclusion, this study has shown that it is possible to achieve very significant accuracies in intrusion detection for IIoT networks. The study has outlined some limitations, thereby suggesting promising areas for future research. These encompass diverse dataset exploration, alternative ML algorithm evaluation, enhanced feature engineering, model optimization, investigation of real-time implementations, and addressing interpretability challenges. Future work should also emphasize scalability and adaptability to accommodate evolving IIoT networks and emerging threat scenarios.

Data availability This study relies on a publicly available dataset (WUSTL-IIOT-2021). This dataset is available from the following reference—Zolanvari, M., Gupta, L., Khan, K. M., & Jain, R. (2021), WUSTL-IIOT-2O2l Dataset for IIoT Cybersecurity Research, Washington University in St. Louis, USA.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval The authors would like to convey their thanks and appreciation to the "University of Sharjah" for supporting this work.

Informed consent This study does not involve any experiments on animals.

References

1. Stouffer K, Pillitteri V, Lightman S, et al (2015) Guide to industrial control systems (ICS) security NIST special publication 800–82 revision 2, pp 1–157
2. Smadi AA, Ajao BT, Johnson BK et al (2021) A comprehensive survey on cyber-physical smart grid testbed architectures: requirements and challenges. Electronics 10:1043. https://doi.org/10.3390/electronics10091043
3. Bonetto R, Sychev I, Zhdanenko O, et al (2020) Smart grids for smarter cities. In: 2020 IEEE 17th annual consumer communications and networking conference (CCNC). https://doi.org/10.1109/CCNC46108.2020.9045309
4. Attar H (2023) Joint IoT/ML platforms for smart societies and environments: a review on multimodal information-based learning for safety and security. J Data Inf Qual. https://doi.org/10.1145/3603713
5. Calabretta M, Pecori R, Vecchio M, Veltri L (2018) MQTT-AUTH: a token-based solution to endow MQTT with authentication and authorization capabilities. J Commun Softw Syst 14:320–331. https://doi.org/10.24138/jcomss.v14i4.604
6. Calabretta M, Pecori R, Veltri L (2018) A token-based protocol for securing MQTT communications. In: Proceedings of the 26th international conference on software, telecommunications and computer networks, SoftCOM 2018, pp 373–378. https://doi.org/10.23919/SOFTCOM.2018.8555834
7. Nti IK, Adekoya AF, Narko-Boateng O, Somanathan AR (2022) Stacknet based decision fusion classifier for network intrusion detection. Int Arab J Inf Technol 19:478–490. https://doi.org/10.34028/iajit/19/3A/8
8. Abdul Rahman Al-chikh Omar A, Soudan B, Ala' Altaweel (2023) A comprehensive survey on detection of sinkhole attack in routing over low power and Lossy network for internet of things. Internet Things (Netherlands). https://doi.org/10.1016/j.iot.2023.100750
9. Samara G, Aljaidi M, Alazaidah R, et al (2023) A comprehensive review of machine learning-based intrusion detection techniques for IoT networks. In: Artificial intelligence, Internet of Things, and society 5.0. pp 465–473
10. Manderna A, Kumar S, Dohare U et al (2023) Vehicular Network Intrusion Detection Using a Cascaded Deep Learning Approach with Multi-Variant Metaheuristic. Sensors 23:8772. https://doi.org/10.3390/s23218772
11. Alamleh A, Albahri OS, Zaidan AA et al (2023) Federated Learning for IoMT Applications: A Standardization and Benchmarking Framework of Intrusion Detection Systems. IEEE J Biomed Heal Informatics 27:878–887. https://doi.org/10.1109/JBHI.2022.3167256
12. Surakhi O, García A, Jamoos M, Alkhanafseh M (2022) The Intrusion detection system by deep learning methods: issues and challenges. Int Arab J Inf Technol 19:501–513. https://doi.org/10.34028/iajit/19/3A/10
13. Keliris A, Salehghaffari H, Cairl B, et al (2016) Machine learning-based defense against process-aware attacks on industrial control systems. In: Proceedings of 2016 IEEE international test conference (ITC), pp 1–10. https://doi.org/10.1109/TEST.2016.7805855
14. Ullah I, Mahmoud QH (2017) A hybrid model for anomaly-based intrusion detection in SCADA networks. In: Proceedings of 2017 IEEE international conference on big data (big data), pp 2160–2167. https://doi.org/10.1109/BigData.2017.8258164
15. Vulfin AM, Vasilyev VI, Kuharev SN et al (2021) Algorithms for detecting network attacks in an enterprise industrial network based on data mining algorithms. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/2001/1/012004
16. Beaver JM, Borges-Hink RC, Buckner MA (2013) An evaluation of machine learning methods to detect malicious SCADA communications. In: Proceedings of 2013 12th international conference on machine learning and applications ICMLA, vol 2, pp 54–59. https://doi.org/10.1109/ICMLA.2013.105
17. Zhang Y, Ilić MD, Tonguz OK (2011) Mitigating blackouts via smart relays: a machine learning approach. Proc IEEE 99:94–118. https://doi.org/10.1109/JPROC.2010.2072970
18. Maglaras LA, Jiang J (2014) Intrusion detection in SCADA systems using machine learning techniques. In: Proceedings of 2014 science and information conference, pp 626–631. https://doi.org/10.1109/SAI.2014.6918252
19. Song Y, Luo W, Li J, et al (2021) SDN-based Industrial Internet Security Gateway. In: 2021 International conference on security, pattern analysis, and cybernetics (SPAC), pp 238–243. https://doi.org/10.1109/SPAC53836.2021.9539961
20. Zolanvari M, Teixeira MA, Gupta L et al (2019) Machine learning-based network vulnerability analysis of industrial Internet of Things. IEEE Internet Things J 6:6822–6834. https://doi.org/10.1109/JIOT.2019.2912022
21. Teixeira MA, Gupta L, Khan KM, Machine RJ (2021) WUSTL-IIOT-2021 dataset for IIoT cybersecurity research. Washington University, St. Louis
22. Siebert J, Joeckel L, Heidrich J et al (2022) Construction of a quality model for machine learning systems. Softw Qual J 30:307–335. https://doi.org/10.1007/s11219-021-09557-y
23. Sarker IH (2021) Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci. https://doi.org/10.1007/s42979-021-00815-1
24. Eid AM, Nassif AB, Soudan B, Injadat MN (2023) IIoT network intrusion detection using machine learning. In: 2023 6th International conference on intelligent robotics and control engineering (IRCE). IEEE, pp 196–201
25. Ting KM (1998) Inducing cost-sensitive trees via instance weighting. Lect Notes Comput Sci (Subser Lect Notes Artif Intell Lect Notes Bioinf) 1510:139–147. https://doi.org/10.1007/bfb0094814
26. Zhang YP, Zhang LN, Wang YC (2010) Cluster-based majority under-sampling approaches for class imbalance learning. In: Proceedings of 2010 2nd IEEE international conference on information and financial engineering, pp 400–404. https://doi.org/10.1109/ICIFE.2010.5609385
27. Richman R, Wuthrich MV (2020) Nagging predictors. SSRN Electron J. https://doi.org/10.2139/ssrn.3627163
28. Mesevage TG (2021) Data cleaning steps and process to prep your data for success. MonkeyLearn, Montevideo
29. Tableau (2022) Data cleaning: definition, benefits, and how-to. Tableau, Mountain View
30. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. https://doi.org/10.1186/s12864-019-6413-7
31. Chicco D, Jurman G (2023) The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. BioData Min. https://doi.org/10.1186/s13040-023-00322-4

32. Khafajeh H (2020) An efficient intrusion detection approach using light gradient boosting. J Theor Appl Inf Technol 98:825–835