ELAIDS: An Improved Ensemble Learning Approach For Intrusion Detection System in Industrial IoT

Kumar Saurabh[1],V.Vamshi[1],Uphar Singh[1],Rahamatullah Khondoker[2], O.P. Vyas[1]
[1]Department of Information Technology, Indian Institute of Information Technology Allahabad, Prayagraj, India
[2]Department of Business Informatics, THM University of Applied Sciences, Friedberg, Germany
pwc2017001@iiita.ac.in,iit2020199@iiita.ac.in,pse2017003@iiita.ac.in, rahamatullah.khondoker@mnd.thm.de, opvyas@iiita.ac.in

***Abstract*—The Industrial Internet of Things (IIoT) has revolutionised the industrial sector by establishing connectivity between machines and devices through the internet. The heightened level of connectivity has concomitantly amplified the vulnerability to cyber-attacks, which pose a threat to the uninterrupted functioning of organisations leads to the economic and reputational damages for businesses, as well as the misappropriation of confidential data. Machine Learning (ML) based Network Intrusion Detection Systems (NIDS) have gained notable popularity to safeguard against security breaches by monitoring anomalous behaviours. The feature sets of the presently accessible NIDS datasets exhibit significant differences which significantly influences the performance of ML models making it unreliable. The proposed approach is based on the NetFlow-based (NF-ToN-IoT-v2) dataset, which is a standardised feature set for NIDS datasets. The primary contribution of this study lies in its capacity to systematically and dependably assess the effectiveness of Ensemble and ML-based traffic classifiers across a diverse array of network topologies, attacks, and other relevant factors. The experimental findings indicate that XG-Boost exhibited a maximum accuracy of 96.01% in detecting various forms of attacks in an IIoT network.**

***Keywords*—IIoT, Intrusion Detection System, Cyber security, Attack Detection**

I. Introduction

The Industrial Internet of Things (IIoT) pertains to an interconnected network of intelligent devices, including connected devices and sensors, that are utilized to establish systems in the manufacturing industry [1]. These systems are utilized in industrial environments to track, gather, share, and analyze data and communicate with other devices and systems. By the year 2025, it is estimated that the Internet will connect about 35 billion things [2]. With more and more devices and systems being connected through the internet and other networks, there are increased risks of cyber-attacks, data breaches, and other security threats [3]. As technology continues to advance, the threat span and attack surfaces are also growing with it (Fig. 1). Therefore security measures for IIoT devices are continuously evolving and improving with time, but there are still concerns regarding their effectiveness [3].

So in order to preserve the security of systems it is required to have a better version of security such as Network Intrusion detection systems(NIDS), which are utilized for identifying cyber-attacks [4]. NIDS [5] is useful for evaluating the traffic of incoming networks. After the analysis of traffic in the entire subnet, if an attack is identified, an alert signal is generated. The implementation of Network Intrusion Detection Systems (NIDS) is of paramount importance in safeguarding computer networks from unauthorized entry and malevolent actions. Conventional rule-based network intrusion detection systems (NIDS) encounter difficulties in identifying complex and dynamic cyber attacks. In recent years, there has been a notable surge in the utilization of machine learning methodologies to improve the performance of NIDS. Researchers have investigated the application of machine learning algorithms for the purpose of inspecting network traffic patterns and detecting possible intrusions. Nevertheless, there exists a need for research in enhancing the precision and effectiveness of Network Intrusion Detection Systems (NIDS) through the utilization of machine learning methodologies. A promising way involves the implementation of ensemble learning, a technique that combines numerous machine-learning models to attain superior detection performance of NIDS. Researchers have investigated the application of machine learning algorithms for the purpose of inspecting network traffic patterns and detecting possible intrusions. Nevertheless, there exists a need for research in enhancing the precision and effectiveness of Network Intrusion Detection Systems (NIDS) through the utilization of machine learning methodologies. Despite of the better performance of ensemble learning approaches, further research is required to optimize ensemble techniques and develop novel ensemble architectures to build reliable NIDSs.

Additionally, the lack of labeled and diverse datasets with updated attack feature in Industrial Control System(ICS) systems environments are also one of the biggest research challenges for training and evaluating machine learning-based NIDS remains a challenge, requiring further efforts to collect and curate representative datasets for comprehensive benchmarking and comparative analysis. Addressing these research gaps will contribute to the development of more robust and effective network intrusion detection systems using ensemble machine learning approaches.

The study introduces an NIDS using ensemble learning algorithms to detect cyber-attacks on IIoT devices. Tree-based models, including Extra Trees, Random Forest, CatBoost, and XGBoost, were used and compared with base machine

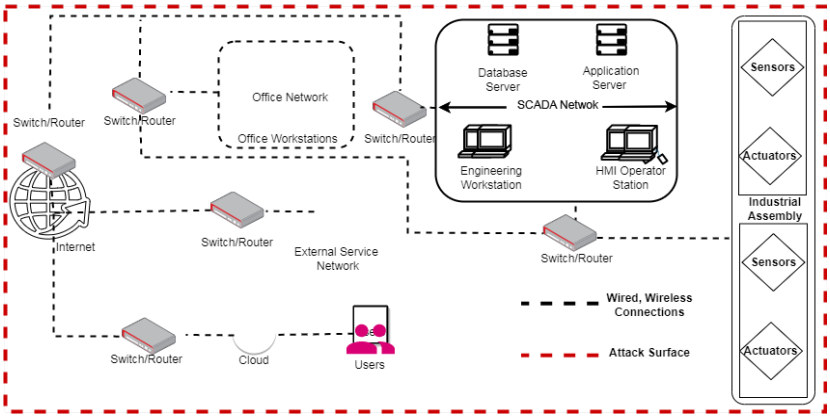

Fig. 1. Attack Surface in an Industrial IoT

learning models like Decision Tree(DT), K- nearest neighbor(KNN) and support vector machine (SVM).for diverse attack categories detection. This work demonstrates the efficacy of machine learning in enhancing IIoT security, offering real-time detection capabilities and effective feature selection using tree-based models. The dataset used in this manuscript is a publicly available dataset consisting of network traffic traces collected from a realistic IoT network environment.

The remaining sections of this study are structured as follows: Section II describes literature reviews, Section III describes the dataset description, Section IV includes our proposed methodology, Section V contains the results and performance, and Section VI gives the conclusion and future perspectives.

II. Literature Review

A. State-of-The-Art

In article [6], the authors are describing the cyber-attacks on autonomous vehicles and connected vehicles. The authors proposed an Intrusion Detection System (IDS) for attack detection to secure vehicular networks. They introduced an IDS based on tree-based ML models to detect threats on vehicular networks. For the class imbalance problem, SMOTE oversampling method was applied and a tree-based averaging feature selection approach was utilized to remove irrelevant features. The IDS was tested on 2 datasets namely the CAN-intrusion dataset and CICIDS2017. The accuracy of CAN-intrusion is 99.98% and CICIDS2017 is 99.86%. In article [7], the authors proposed an IDS for vehicular networks using the dataset ToN-IoT.The dataset has missing values and class imbalance problems. The combination of Chi2 and SMOTE is used to feature selection and class imbalance problems respectively after the preprocessing. Among all the proposed models XGBoost works well for binary and multi-class classification problems with an accuracy of 97% and 98% respectively. In article [8], the authors are describing improving security in the Internet of Vehicles(IoV). The authors proposed a method called LCCDE(Leader Class and Confidence Decision Ensemble). LCCDE identifies the best Machine Learning models for the detection of attacks and combines them into an ensemble model. This method selects the best machine learning models for the detection of an attack and combines them into an ensemble model. XGBoost, CatBoost, and LightGBM are the models used in this paper. This model has achieved an F1-score of 99.8% on the CICIDS2017 dataset.
In article [9], the authors analyzed the UNSW-NB15 dataset and employed ensemble learning methods by using the stacking technique. The authors combined Random forest, Naive Bayes, and SVM and used logistic regression as a meta-classifier. The approach resulted in achieving an accuracy of 95%. In article [10], the authors suggested a model for an intrusion detection system called a multi-tiered hybrid intrusion detection system (MTH-IDS), which can identify various types of zero-day cyber-attacks and known attacks on vehicular networks. It consists of four tiers of learning models. The first tier consists of tree-based supervised learners. The second tier consists of Bayesian optimization with a Tree-structured Parzen estimator. The third tier consists of the CL-K-means model. The fourth tier consists of Bayesian optimization with Gaussian Processes. The combination of these four tiers has improved the performance of the model in achieving an accuracy of 99.8% and 99.9% on CICIDS2017 and CAN-intrusion respectively.

B. Datset for Industrial IoT

To build a ML based NIDS many publicly available datasets like ToN-IoT, UNSW-NB15, CICIDS 2017, CICIDS 2018, CICIDS2021, NSL-KDD, KDD Cup, etc. have been used for a long time by Researchers. In this manuscript, the NF-ToN-IoT-v2 dataset is a variant of the ToN-IoT dataset but with a NetFlow feature set. NF-ToN-IoT v2 is a NetFlow-based dataset generated from publicly available packet capture (pcap) files of the ToN-IoT dataset [11]. The ToN-IoT dataset is a publicly available heterogeneous dataset consisting of network traffic traces collected from a realistic IoT network environment [12]. To generate the NF-ToN-IoT dataset, pcap files of the dataset ToN-IoT were processed to extract NetFlow records [11], which are a type of network flow data that summarizes information about network traffic, such as the

TABLE I
Comparison of Datasets for Attack Detection in Industrial IoT

Dataset	Description	Features	Size	Attacks	Year	Samples (Millions)
NF ToN-IoT V2	Comprehensive IIoT dataset capturing realistic ICS network traffic.	43	Large	8	2021	4.5
ToN-IoT	IIoT dataset capturing network traffic for intrusion detection research.	30+	Large	5	2018	2.5
UNSW-NB15	Network traffic dataset from a controlled environment.	42	Large	9	2015	2.5
CICIDS2017	Real network traffic dataset from an enterprise environment.	80+	Large	16	2017	2.8
CICIDS2018	Network traffic dataset with synthetic attacks in an enterprise environment.	80+	Large	23	2018	2.8
CICIDS2021	Network traffic dataset capturing real and simulated attacks in an enterprise environment.	78	Large	15	2021	4.4
NSL-KDD	Network traffic dataset derived from the original KDD Cup 1999 dataset.	41	Moderate	4	2009	1.5
KDD Cup	Network traffic dataset for intrusion detection in computer networks.	41	Large	22	1999	4.9

source and destination IP addresses, port numbers, protocol type, and several packets and bytes. The NetFlow-derived datasets ensures more concise and organized depiction of the network traffic data. This can prove advantageous for a range of undertakings, including network surveillance, identification of aberrations, and classification based on machine learning. The aforementioned dataset possesses versatility in its applicability, as it can be utilized for a multitude of purposes, including but not limited to the development and assessment of intrusion detection systems, evaluation of security solution efficacy, and the training of machine learning models for network traffic classification. Table I lists the popular datasets used for NIDS.

C. Summery & Research Gap

The literature review shows that ensemble learning techniques improve NIDS performance for vehicular networks and IoT, but there is no study on ensemble learning-based NIDS for Industrial IoT contexts. LCCDE and multi-tiered hybrid IDS are used in certain studies, however, IIoT applications and evaluations are scarce. The literature mostly uses CICIDS2017 and CAN-intrusion datasets, which may not completely depict the complexity and diversity of Industrial IoT threats. This research gap is listed below:

- To improve NIDS accuracy, robustness, and real-time capabilities in IIoT contexts, ensemble strategies such as integrating various machine learning models or using diverse feature selection methods need more study.
- To create and use realistic and large-scale datasets that capture the specific characteristics and problems of IIoT networks to evaluate and benchmark ensemble learning-based NIDS. Addressing these research gaps would help build better ensemble learning-based NIDS for Industrial IoT, improving industrial security and threat detection.

III. Proposed NIDS Framework

After splitting the dataset into "Training Set" and "Testing Set", Data Transformation is the first step that deals with the class imbalance problem which helps to increase the classification accuracy followed by Label Encoding, Data Normalization, Data Augmentation, and Feature Engineering to improve training efficiency. The steps involved in this methodology are explained below.

A. Dataset

The NF-ToN-IoT v2 dataset comprises approx16 million data records. Out of the total data records, 63.99% or approx 10 millions samples are classified as attack data, whereas the remaining 36.01% or approx 6 millions samples are benign.

B. Data Preprocessing

1) Data transformation:

a) Data transformation using clustering: Training machine learning models on network traffic datasets is a challenging task that demands substantial time and computational resources [10]. The process of selecting a smaller subset of data from a larger dataset, known as sampling, is a commonly employed technique to reduce the complexity of model training [10]. The proposed methodology utilizes a clustering-based approach employing the k-means algorithm to extract a subset of data that exhibits high representativeness. Cluster sampling is a frequently employed method for the purpose of selecting a subset of data for analysis [10].

This technique involves grouping the initial data points into multiple clusters and then randomly selecting a proportion of data from each cluster to generate a representative subset. K-means is a commonly used unsupervised learning algorithm in machine learning, which involves clustering a group of data points based on their similarities [13]. The k-means algorithm initiates by randomly selecting K centroids and assigning each data point to the centroid nearer to it. The centroids are then shifted to the mean of all the data points assigned to that particular cluster. This procedure is repeated until either the centroids stop moving or a predefined maximum number of iterations is reached. The primary goal of the k-means algorithm is to reduce the total sum of squared distances between each data point and its corresponding cluster centroid, denoted by [10]:

$$\sum_{i=0}^{n_k} \min_{u_j \in C_k} (\mathbf{x}_i - u_j)^2$$

where $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ is the data matrix; The centroid or mean of the cluster C_K is denoted by u_j and is computed as the average of all samples in C_k where n_k is the total number of sample points in C_k. The time complexity of k-means is $O(nkt)$, where n is the size of the data, k is the no. of clusters, and t is the total iterations [13].

To divide the data points into k clusters based on their distance, a k-means clustering method is used. The distance

TABLE II
The attacks with the distributions found in the NF-ToN-IoT v2 dataset :

S.No.	Class	Count	Description
1	Benign	6099469	Normal unmalicious flows
2	Backdoor	16809	A method for attacking remote systems by responding to specially created client programs
3	DoS	712609	A deliberate attempt to overtax a computer system resources to restrict access to or make data unavailable
4	DDoS	2026234	A DoS-like attack that uses a variety of dispersed sources
5	Injection	684465	many attacks try to change how code is executed by providing untrusted inputs; two of the most common are SQL injections and code injections.
6	MITM	7723	A technique known as "Man in the Middle" involves inserting an attacker between a victim and the host with the victim is attempting to contact in order to eavesdrop on traffic and communications.
7	Password	1153323	Includes several attacks that use brute force or sniffer to recover passwords.
8	Ransomware	3425	A cyberattack that encrypts the data kept on a host and demands payment in exchange for the method or key used to decrypt it.
9	Scanning	3781419	An assortment of methods together referred to as probing that seek to learn details about hosts and networks
10	XSS	2455020	A cross-site scripting attack using online applications to send harmful scripts to end users

is computed using Manhattan or Euclidean or Mahalanobis metrics. The process involves applying k-means clustering to the original dataset to group the data samples into k clusters, followed by random sampling of a certain percentage of data from each cluster to create a representative subset. The percentage of data selected for sampling depends on the resources available.

b) Label Encoding: In machine learning, encoding is a technique that is used to transform categorical data into numerical data. This is done to enable algorithms to better interpret and process the data. Since numerous machine learning algorithms only accept numerical input data, categorical data must be transformed for the algorithms to make predictions. Label encoding is one such technique, where each distinct category is assigned a numerical label, and each label denotes a unique category. The labels are allocated in an arbitrary fashion with no inherent order or ranking between them.

2) Data Normalization : Network traffic data usually consists of features with significantly varying ranges. To enhance the performance of machine learning models trained on such data, it is recommended to normalize the dataset [14] . If the feature scales are largely different and the dataset is not normalized, a biased model may result, prioritizing the significance of large-scale features. To address this, the Z-score technique is used to normalize the features. This technique transforms the data into a distribution with a mean of 0 and a standard deviation of 1 by subtracting the mean value of each feature from its respective data points and dividing it by the feature's standard deviation. This process guarantees that the data is on a uniform scale and can be used effectively in machine learning models. When using the Z-score method to normalize features, each normalized feature value, denoted by x_n, is determined as follows:

$$x_n = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the value of feature taken, μ is the mean and σ is the standard deviation respectively [10].

3) Data Augmentation: Imbalanced classes are a common problem in network traffic data, which can lead to biased models [15]. Resampling methods are often used to address this issue. One such technique is SMOTETomek, which is a hybrid approach that combines two sampling techniques, SMOTE and Tomek links, to balance the minority and majority classes in the dataset. SMOTE is used to oversample the minority class by creating synthetic samples that are similar to existing samples in that class, thereby avoiding a bias towards the majority class. On the other hand, Tomek links are pairs of samples in different classes that are close to each other but are classified differently. Removing these pairs can help to improve the separation between the classes and reduce the overlap. The SMOTETomek technique first oversamples the minority class using SMOTE and then removes majority class samples identified as Tomek links. This approach helps to improve the balance between the classes and reduces the risk of overfitting or underfitting the minority class.

C. Feature Engineering

For training machine learning models, feature selection methods such as Information Gain are used to remove irrelevant features while retaining essential features. Information Gain (IG) is a method that measures the amount of information a feature can bring to the target variable by calculating the changes in entropy. In the proposed system, Information Gain is preferred due to its fast speed and low computational complexity of O(n) in obtaining an importance score for each feature. Mutual information calculates the statistical dependence between two variables and is the name given to information gain when we are using it for the selection of features. The Mutual Information score is used to evaluate the relationship between two variables and is determined by measuring the reduction in uncertainty. A higher score indicates a stronger association between the two variables, and the score is always equal to or greater than zero. The Information Gain value associated with a particular feature X and target variable T is represented by $IG(T|X)$

Information Gain value using feature X denoted by [10]:

$$IG(T \mid X) = H(T) - H(T \mid X), \tag{2}$$

where $H(T)$ is entropy T, $H(T|X)$ is conditional entropy of T given X. If the value of $IG(T|X)$ is higher than the value of $IG(T|Y)$, then feature X is considered more significant to the target T than feature Y [10].

TABLE III
Data preprocessing and feature engineering methods and their impact on performance:

Stage	Method	Description	Impact on Performance
Data Preprocessing	K-means Cluster sampling	As network datasets are large, sampling using K-means will generate a subset of data that is highly representative.	Improves training efficiency
	SMOTETomek	Combination of oversampling and under-sampling using SMOTE and Tomek Links. Deals with the class imbalance problem to avoid a biased model.	Handles class imbalance problem
	Label Encoding	Conversion of categorical variables into numerical variables.	Enables transformation of categorical data
	Z-score normalization	Normalization of features to a similar scale and handling outliers.	Increases accuracy of the model
Feature Engineering	Information Gain	Removal of irrelevant features	Improves training efficiency
IDS Designing	XGBoost, ET, RF, CatBoost	Tree-based machine learning algorithms perform well compared to other ML methods on complex data	To detect the attacks

D. *Ensemble and Standard Machine Learning Models:*

The system's proposed methodology involves the selection of tree-based Machine Learning algorithms which includes Random forest(RF), Extra Trees(ET), Extreme gradient boosting (XGBoost), and CatBoost. Apart from the proposed algorithms, we are also performing evaluation using the common classification models such as Decision Tree, K-nearest neighbor (KNN), and support vector machine (SVM).

1) Decision Tree: The decision tree is a prevalent machine learning model that employs a tree structure based on the divide and conquers strategy to accomplish classification tasks. The recursive construction of this structure involves the systematic selection of the optimal feature to partition at each node, utilizing a criterion such as entropy or information gain. The Decision Tree serves as the fundamental component of the individual trees that are aggregated in the ensemble. The preceding algorithms are constructed based on the concept of decision trees.

2) Random Forest: Random forest (RF) [16] is a type of ensemble learning classifier that utilizes the majority voting rule. This involves the creation of multiple decision trees, and the final classification result is determined by a majority voting rule among all the decision trees. The use of multiple decision trees is intended to increase the accuracy and robustness of the system's predictions. In a random forest algorithm, each decision tree is trained using a randomly selected subset of the training data and a random subset of features. The predictions made by each individual tree are then combined to make the final prediction.

3) Extra Trees: Extra Trees [17] is a type of ensemble model that is similar to Random Forests as it also uses multiple decision trees to make predictions. However, the way Extra Trees constructs these trees is different. In Extra Trees, the splitting thresholds for each feature are randomly chosen instead of selecting the best-split point, which adds more randomness and reduces the model's variance. This method is effective for high-dimensional data with noisy features and is relatively fast to train.

4) XGBoost: XGBoost [18] is also an ensemble learning algorithm it is designed to improve both the speed and performance of the model through the use of gradient descent. This algorithm combines multiple decision trees to create an effective ensemble model. XGBoost algorithm utilizes gradient boosting to enhance the accuracy of the decision trees. The algorithm works by creating a series of decision trees where each subsequent tree is built to minimize the loss function with regard to the previous tree's predictions. The minimization is done using a gradient descent approach.XGBoost has a complexity of $O(Kd\|\mathbf{x}\|\log n)$, where d is the maximum depth of the tree, K is the no of trees used,n is the size of the data and $\|\mathbf{x}\|$ is non-missing samples.

5) CatBoost: CatBoost [19] is a powerful open-source gradient boosting algorithm that uses decision trees and is specialized in handling categorical data. It has shown impressive performance in various machine learning tasks, especially those with high-dimensional categorical data. The algorithm uses an ordered boosting approach to improve the accuracy of decision trees by training them in a specific order. It also has built-in features to prevent overfitting, such as L2 regularization.CatBoost model has a complexity of $O(SN)$, where N is the no of base Decision Tree models and S is the no of possible permutations of the subsets used to train each base model.

If no. of instances are n, no.of features are f and no .of decision trees are t the time complexity of Extra trees, Random forest, and Decision Trees are $O(nft)$, $O\left(n^2\sqrt{f}t\right)$ and $O\left(n^2 f\right)$ [6].

The algorithm choosing reasons are as follows [6]:
1. For non-linear and complex datasets like network traffic datasets these ensemble algorithms are much more efficient to use. This makes them more effective than other machine learning models such as linear models.
2. The models calculate feature importance while training, which helps with feature engineering procedures.
3. Tree-based algorithms have a random construction process, which results in the creation of diverse decision trees. This diversity helps to develop a robust ensemble model that can generalize better compared to other machine learning algorithms.

To achieve better performance on the proposed model we have to find the best hyper-parameters.In order to get the best hyper-parameters we are performing a grid search

with cross-validation on the proposed models. The process involves defining a range of hyper-parameters in the form of a grid, which is a set of all possible combinations of hyper-parameters. Subsequently, the classifier is trained and assessed through cross-validation for each hyper-parameter combination in the grid. The optimal hyper-parameters for the model are determined by selecting the hyper-parameters that yield the highest cross-validation score.This process confers benefits in terms of identifying optimal hyper-parameters for the classifier, thereby enhancing the accuracy and generalizability of the model. Furthermore, it streamlines the hyper-parameter tuning process, which can prove to be a tedious and intricate task when performed manually.

IV. Results and Performance Analysis

A. Experimental Setup

The experiments were carried out on a system. The specifications were Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 16GB RAM. The version of Python installed was 3.11.2.

B. Validation Metrics

To evaluate the performance of machine learning models several metrics have been used :

1) True Positives (TP): The number of observations that were correctly classified as positive by the ML model.

2) False Positives (FP): The number of observations that were classified as positive by the ML model, but were actually negative.

3) False Negatives (FN): The number of observations that were classified as negative by the ML model, but were actually positive.

4) True Negatives (TN): The number of observations that were correctly classified as negative by the ML model.

5) Accuracy: The proportion of correctly classified instances to the total number of instances.

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

6) Precision: It is the ratio of true positives to the total number of observations classified as positive.

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

7) (Recall/Sensitivity): It is the ratio of true positives to the total number of positive instances.

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

8) F1 Score: A weighted average of precision and recall that takes both metrics into account.

$$F1 \text{ Score } = \frac{2 * \text{ Recall } * \text{ Precision}}{\text{Recall } + \text{ Precision}}$$

These metrics can help us evaluate the performance of a classification model and determine if it is suitable for our needs.

C. Performance Analysis

In this study, a comparative study has been performed to analyze how well various machine learning models performed on the NF-ToN-IoT-v2 dataset with the aim of predicting the target label, which is considered as a multi-classification problem. We trained and tested different ensemble approaches in Machine Learning models like XGBoost, Random Forest, Extra Trees, and CatBoost. Apart from the proposed models we have also implemented other base ML models such as DT, KNN, and SVM.

To evaluate the performance of the models, 5-fold cross-validation is also used to measure reliable accuracy, precision, recall, and F1 scores. Results tabulated in Table IV shows that the XGBoost model achieved the highest accuracy of 96%, other models such as Extra Trees, Random Forest, and CatBoost also performed well, achieving an accuracy of 95%, 94 %, and 93% respectively. Apart from the ensemble models, other models such as DT, SVM, and KNN also produced good results with an accuracy of 94%, 94%, and 93% respectively. For XGBoost, the multi-classification results are also tabulated in Table V. It is clearly visible that the model is good at detecting all attacks but struggles with Dos attack detection.

Overall, the results suggest that Random Forest and XG-Boost are the most suitable models for predicting the target variable on the dataset of NF-ToN-IoT-v2. However, it is important to note that further optimization and tuning of the models may lead to even better performance.

TABLE IV
Model performance on evaluation metrics

Model	Accuracy	Precision	Recall	F1-score
XGBoost	**96.01**	**96.51**	**96.10**	**96.32**
Extra Trees	95.23	96.56	95.23	95.80
Random Forest	95.07	96.76	95.07	95.72
CatBoost	93.77	96.35	93.77	94.78
Decision Tree	0.75	0.76	0.71	0.73
SVM	0.89	0.91	0.87	0.88
KNN	93.23	0.79	0.74	0.76

TABLE V
XG-Boost Multiclassification performance

S.No.	Attacks	Precision	Recall	F1- score
1	Benign	.95	.95	.95
2	Backdoor	1.00	1.00	1.00
3	DDos	.96	97	.96
4	Dos	.16	.26	.20
5	Injection	.65	0.74	0.69
6	MITM	0.96	0.92	0.93
7	Password	.90	0.96	0.94
8	Ransomware	1.00	1.00	1.00
9	Scanning	.93	0.92	0.93
10	Xss	.77	0.88	0.82
	Macro Avg.	0.83	0.86	0.84
	Weighted Avg.	0.97	0.96	0.96

V. Conclusion

The present study introduces an Intrusion Detection System (IDS) that employs machine learning algorithms for the purpose of identifying attacks on Industrial Internet of Things (IIoT) devices. The study utilized tree-based models, namely Extra Trees, Random Forest, CatBoost, and XGBoost, to attain a notable level of precision in the classification of diverse attack categories. The experimental outcomes indicate that

the proposed Intrusion Detection System (IDS) attained a maximum accuracy of 96.01% in identifying various forms of attacks on Industrial Internet of Things (IIoT) devices.

Future work can explore the use of federated learning techniques for the proposed IDS, which can potentially address privacy concerns and enable collaborative learning across multiple IIoT devices. Additionally, the deployment of the IDS on a real-world IIoT environment can provide valuable insights into its practical effectiveness and performance.

REFERENCES

[1] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in industry*, vol. 101, pp. 1–12, 2018.

[2] L. Fetahu, A. Maraj, and A. Havolli, "Internet of things (iot) benefits, future perspective, and implementation challenges," in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2022, pp. 399–404.

[3] X. Yu and H. Guo, "A survey on iiot security," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 2019, pp. 1–5.

[4] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[5] K. Shiomoto, "Network intrusion detection system based on an adversarial auto-encoder with few labeled training samples," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 5, 2023.

[6] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in internet of vehicles," in *2019 IEEE global communications conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[7] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset," *IEEE Access*, vol. 9, pp. 142 206–142 217, 2021.

[8] L. Yang, A. Shami, G. Stevens, and S. De Rusett, "Lccde: A decision-based ensemble framework for intrusion detection in the internet of vehicles," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 3545–3550.

[9] M. Abirami, U. Yash, and S. Singh, "Building an ensemble learning based algorithm for improving intrusion detection system," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 2020, pp. 635–649.

[10] L. Yang, A. Moubayed, and A. Shami, "Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2021.

[11] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile networks and applications*, pp. 1–14, 2022.

[12] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton˙iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[13] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *2010 Third International Symposium on intelligent information technology and security informatics*. Ieee, 2010, pp. 63–67.

[14] K. M. Ali Alheeti and K. McDonald-Maier, "Intelligent intrusion detection in external communication systems for autonomous vehicles," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 48–56, 2018.

[15] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, vol. 433, pp. 346–364, 2018.

[16] A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with smote and feature reduction," in *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*. IEEE, 2013, pp. 127–132.

[17] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, pp. 3–42, 2006.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[19] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.