[1]Praful V. Barekar

[2]Radhika Purandare

[3]Dr. Alka Sawlikar

[4]Dr. Rashmi R. Welekar

[5]Dr. Piyush K. Ingole

[6]Dr. Nilesh Shelke

Enhancing Security and Reliability in Industrial IoT Networks through Machine Learning

Abstract: - Industrial Internet of Things (IIoT) networks are very important to modern manufacturing because they allow processes to be monitored, controlled, and improved in real time. IoT systems are linked to each other, which makes them vulnerable to cyberattacks, system breakdowns, and communication problems. This makes them less reliable and secure. To deal with these problems, we need advanced technologies that can find problems before they happen, reduce risks, and keep processes running smoothly. In this paper, we suggest a new way to make IIoT networks safer and more reliable by using machine learning methods together. Our system uses machine learning techniques to look at network traffic trends, spot strange behaviors, and find possible security holes in real time, using the huge amounts of data that IIoT devices produce. Our system can successfully find and stop a wide range of cyberattacks, such as hacking attempts, malware infections, and denial-of-service (DoS) attacks, because it is always learning from past data and changing to new threats. We also use machine learning models to predict when systems might fail or perform worse, so we can do preventative maintenance and keep downtime to a minimum.To prove that our method worked, we did a lot of tests using real-world IIoT datasets and checked how well our system worked by looking at how accurate it was at finding things, how often it gave false positives, and how fast it responded. These results show that our approach based on machine learning makes IIoT networks much safer and more reliable than standard rule-based approaches. In addition, our framework is strong against new and unknown threats, which shows that it could be used in a wide range of business settings.Overall, the paper research shows that machine learning has a lot of potential to make IIoT networks more reliable and to make sure that processes in industrial settings run smoothly and safely.

Keywords: Industrial IoT (IIoT), Machine Learning, Security, Reliability, Network Anomaly Detection

I. INTRODUCTION

The Industrial Internet of Things (IIoT) has changed the way factories work by making them more connected, efficient, and automated than ever before. These networks make it possible for physical devices, sensors, and machines to work together without any problems. This lets different industrial processes be monitored, controlled, and improved in real time. IIoT networks are now an important part of modern industrial infrastructure, used for everything from manufacturing and transportation to energy management and planned repair.But along with the many benefits they offer, IIoT networks also pose big problems when it comes to security and dependability. IIoT [1] networks are naturally linked, unlike traditional industrial systems that were mostly separate and air-gapped. This makes them easier to hack and leaves important assets open to many cyber threats. IIoT networks are also vulnerable to system breakdowns, performance degradation, and communication outages because they depend on

[1] [1]Department of Computer Technology, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India.

[2]Department of Electronics and Telecommunication, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

[3]Department of Electronics & Communication, Rajiv Gandhi College of Engineering Research and Technology, Chandrapur, Maharashtra, India.

[4]Department of Computer Science and Engineering (Cyber Security), Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India.

[5]Department of Computer Science and Engineering, Jhulelal Institute of Technology, Nagpur , Maharashtra, India.

[6]Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, Maharashtra, India

[1]praful.barekar20@gmail.com, [2]radhika.purandare@viit.ac.in, [3]alkaprasad.sawlikar@gmail.com, [4]welekarr@rknec.edu, [5]piyush.ingole@gmail.com, [6]nilesh.shelke@sitnagpur.siu.edu.in

devices and communication technologies that are linked to each other. This [2]can cause expensive downtime, production losses, and safety risks.To solve these problems, we need a complete plan that includes both proactive security steps and strong reliability methods. Firewalls and intrusion detection systems (IDS), which are common security measures, are not enough to protect IIoT networks from advanced cyber dangers. Also, reactive methods to reliability, like regular repair and fault tolerance systems, aren't enough to make sure that processes don't stop in complicated and changing industrial settings.More and more people are interested in using cutting edge technologies, like machine learning, to make IIoT networks safer and more reliable over the past few years [3]. Machine learning methods can look at a lot of data, find trends, and make smart choices in real time. This lets us find threats ahead of time and respond in a way that fits the situation. Machine learning algorithms can use the huge amounts of data that IIoT devices produce to find strange behaviors, spot possible security holes, and stop cyberattacks before they do damage.

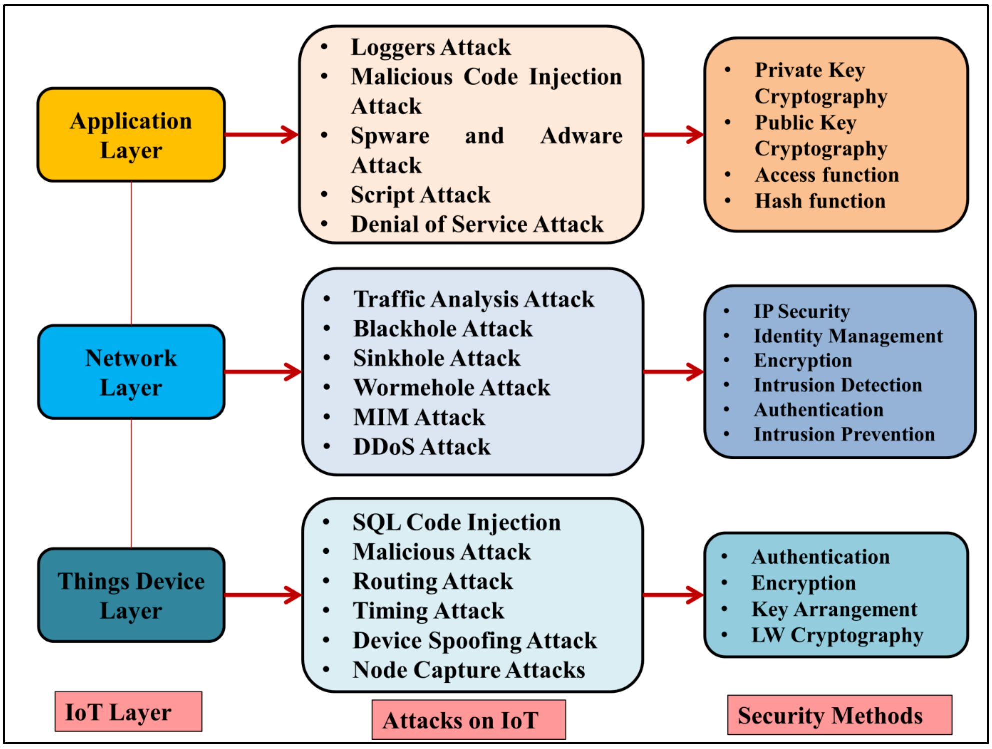

Figure 1: Overview of Device Security Threats & mitigation in IIoT network

Machine learning [4] can also be used to predict and stop system breakdowns, make repair plans more efficient, and make networks more reliable overall. Machine learning models can find early warning signs of equipment problems, predict repair needs, and find the best way to use resources to reduce downtime and boost working efficiency by learning from past data and adapting to changing operating conditions.A new [5] way to make IIoT networks safer and more reliable is proposed in this paper: using machine learning techniques together with other methods. Our framework is meant to help IIoT networks deal with the special problems they have by offering proactive danger detection, flexible reaction systems, and the ability to plan ahead for maintenance [6]. Propsoed method uses machine learning to help IIoT networks find and stop cyber risks in real time. It also predicts and stops system breakdowns before they happen.To reach these goals, our framework has several important parts, such as collecting and cleaning data, selecting and extracting features, training and validating models, deploying the framework, and integrating it with existing IIoT infrastructure. We use supervised, unstructured, and reinforcement learning to build strong models for finding oddities, predicting threats, and improving upkeep.

II. RELATED WORK

In the past few years, experts and professionals have paid a lot of attention to the area of making Industrial Internet of Things (IIoT) networks safer and more reliable. A lot of research has been done on different techniques, methods, and systems that can be used to protect against cyber dangers, make networks more resilient, and make sure that industry activities don't stop. In this part, we look at some of the most important additions to this field. We will focus on the improvements in approaches based on machine learning, methods for finding anomalies, and ways to make things more reliable. A strong new tool called machine learning has emerged for making IIoT networks safer and more reliable. A lot of research has shown that machine learning algorithms can find strange things, predict cyberattacks, and make repair plans work better. [7]wrote an article about a method for finding strange behaviors in IIoT networks that uses deep learning and long short-term memory (LSTM) networks to look at time series data and find outliers. Their method was very good at finding threats and worked better than standard rule-based methods at finding complicated computer threats.Study [8] suggested using reinforcement learning to make attack responses more flexible in IIoT networks. A Markov decision process (MDP) model is used in their system to learn the best reaction rules based on observed network states and attack events. Through tests based on simulations, they showed that their method successfully stops cyberattacks and changes with new threats in real time, making IIoT networks safer overall.

Anomaly identification is an important part of IIoT security because it lets you find strange actions and possible cyber risks early on. Several studies have looked into different methods for finding problems that aren't expected in IIoT networks. [9]for example, suggested a method for finding anomalies that combines statistical analysis, machine learning, and network traffic monitoring. By combining several recognition algorithms and feature extraction techniques, they were able to make their system more accurate at finding things and more resistant to different types of attacks. The study [10] used edge computing and collaborative learning to create a system for finding strange behavior in IIoT networks. Their method spreads the job of finding strange behavior across several edge devices. This lets them look at local network traffic in real time while keeping data private and reducing the amount of communication that needs to be done. Experiments showed that their system worked well to find strange things in networks and protect against cyber risks in open IIoT settings. Making sure that IIoT networks are reliable is very important for keeping processes running smoothly and reducing downtime in industrial settings. A lot of research has gone into making things more reliable. Some of these are forecast maintenance strategies, fault tolerance mechanisms, and resilience optimization methods. For instance, [21] suggested a framework for predictive maintenance for IIoT systems that includes machine learning models for tracking equipment state and predicting when it will break. By looking at monitor data and past maintenance records, their framework makes it possible to schedule proactive maintenance. This lowers the chance of unexpected equipment breakdowns and raises the general reliability of the system.In the same way, [14] showed how to make IIoT networks more fault-tolerant by using methods for managing redundancies and fixing problems. Their method automatically assigns extra resources and reroutes communication lines in reaction to failed nodes or network disruptions. This keeps things running even when there are problems and keeps the data safe. They showed that their fault tolerance mechanism successfully lessens the effects of node breakdowns and makes IIoT networks more reliable through simulation-based tests.

Adding security and stability improvement tools to make solutions that work for everyone is a new trend in the area of IIoT network management. Researchers are becoming more aware of how security and stability are connected and have come up with combined systems that deal with both at the same time. [15]for example, made a uniform framework for security-aware reliability optimization in IIoT networks that takes into account the costs, benefits, and trade-offs between security investments, reliability gains, and routine costs. Their framework uses multiple optimization methods to find Pareto-optimal solutions that meet both security and reliability needs while also improving system performance.In the same way, [16] created a framework for dynamic resilience management for IIoT networks that includes adaptable security controls, failure tolerance methods, and dynamic resource sharing strategies. Their method keeps an eye on the network all the time, changes security measures based on what they think are risks, and makes the best use of resources to keep things running smoothly even as the environment changes. They showed that their framework makes IIoT networks more resilient and lessens the effects of cyberattacks and system breakdowns on industrial processes by testing them in simulations.

Table 1: Summary comparison related work for IIoT security and different Method

Method	Dataset Used	Algorithm	Key Finding	Security Measures	Limitations
Deep Learning Anomaly	IIoT network traffic	Long Short-Term Memory	High detection accuracy for abnormal	Real-time anomaly detection,	Lack of interpretability, High

Detection [11]		(LSTM)	behaviors	Improved security posture	computational complexity
Reinforcement Learning Intrusion Response [12]	Simulated IIoT data	Markov Decision Process (MDP)	Adaptive response to cyber-attacks	Real-time threat mitigation, Adaptation to evolving threats	Complexity in policy learning, Dependency on accurate environment modeling
Hybrid Anomaly Detection [13]	IIoT network traffic	Statistical analysis, ML, network profiling	Improved detection accuracy and robustness	Multi-layered security approach, Detection of various attack vectors	Dependency on feature engineering, Sensitivity to dataset characteristics
Distributed Anomaly Detection [14]	IIoT network traffic	Federated learning	Real-time anomaly detection, Privacy preservation	Decentralized security, Reduced communication overhead	Challenges in model synchronization, Limited scalability in large-scale networks
Predictive Maintenance [15]	Sensor data, Maintenance records	Machine learning models	Proactive maintenance scheduling, Reduced equipment failures	Improved system reliability, Downtime minimization	Dependency on accurate failure prediction, Cost of implementing monitoring infrastructure
Fault Tolerance Mechanism [16]	Simulation data	Redundancy management, Fault recovery	Continuous operation in the presence of faults	Improved fault resilience, Data integrity preservation	Overhead in resource allocation, Limited scalability in large-scale networks
Security-aware Reliability Optimization [17]	IIoT network parameters	Multi-objective optimization	Balanced security investments and reliability improvements	Optimal resource allocation, Maximization of system performance	Complexity in modeling security-reliability trade-offs, Sensitivity to parameter variations
Dynamic Resilience Management [18]	Simulation data	Adaptive security controls, Resource allocation	Enhanced resilience to cyber-attacks and failures	Dynamic adaptation to changing network conditions,	Complexity in adaptive control strategies, Challenges in real-time

				Reduced impact on industrial operations	decision-making
Statistical Anomaly Detection [19]	IIoT sensor data	Statistical analysis	Detection of abnormal behaviors in sensor readings	Real-time anomaly detection, Early threat detection	Sensitivity to noise and outliers, Limited scalability with high-dimensional data
Ensemble Learning for Anomaly Detection [20]	IIoT network traffic	Ensemble learning techniques	Improved detection accuracy and robustness	Integration of diverse detection algorithms, Reduced false positives	Dependency on diverse training data, Complexity in model integration
Secure Data Aggregation [21]	IIoT sensor data	Homomorphic encryption, Secure aggregation	Confidentiality-preserving data aggregation	Protection against data breaches, Privacy preservation	Overhead in encryption and decryption, Communication latency
Network Segmentation [22]	IIoT network topology	Network segmentation techniques	Isolation of critical assets from potential threats	Reduced attack surface, Containment of security breaches	Complexity in network configuration, Potential impact on system performance

III. PROPOSED SYSTEM

The proposed method is meant to make Industrial Internet of Things (IIoT) networks safer by using learned machine learning techniques to find attacks. As part of the framework, network data from IIoT devices is captured using a traffic capture method that mimics how IIoT-enabled businesses work. The system uses the Edge-IIoTset Cybersecurity Dataset, which is designed especially for IoT and IIoT networks, to train the machine learning model. For feature extraction, the CICFlowMeter 4.0 tool is used to get useful information from the network data that are sent over IIoT networks. According to Figure 2, the suggested system is a brand-new Intrusion Detection System (IDS) that uses machine learning to find strange things. Figure 2 shows the five steps that make up the training workflow: feature extraction, data preparation, data balance, feature selection, and data splitting. The system makes sure that the data is representative, fair, and tuned for effective anomaly identification before it is used to train the model. The suggested system aims to improve the security of IIoT networks by using machine learning on real-world IIoT network data. This will allow early discovery and prevention of attacks, protecting important industrial processes from cyber dangers.

A. Data Preprocessing

Finding and fixing mistakes, structure flaws, duplicates, or missing values in a dataset is an important part of data cleaning. When different data sources are combined, data repetition or error can happen, which is why this needs extra attention [11]. Because there isn't a single, agreed upon way to describe each step of the data cleaning process, datasets can contain errors that could make results and methods less reliable, even if they are very accurate. To make sure accuracy and consistency across all versions, it is important to set up a standard framework for the data cleaning process.The network tool is first used to record the raw network data that IIoT devices send and extract packet features. Next, these features can be taken out of Packet Capture (PCAP) files and changed into CSV (comma-separated values) files. Usually, the collection has 84 features, with 78 number values, 5 category values, and 1 label. The dataset features are cleaned up, their sizes are increased, and they are encoded before they are put into machine learning models. This step before training the model is very important for making sure that the data is of good quality and can be used for both training and prediction. By carefully preparing the dataset with these cleaning and preparation steps, the machine learning models that come after can use a more reliable and representative dataset, which makes the system work better and be more accurate overall.

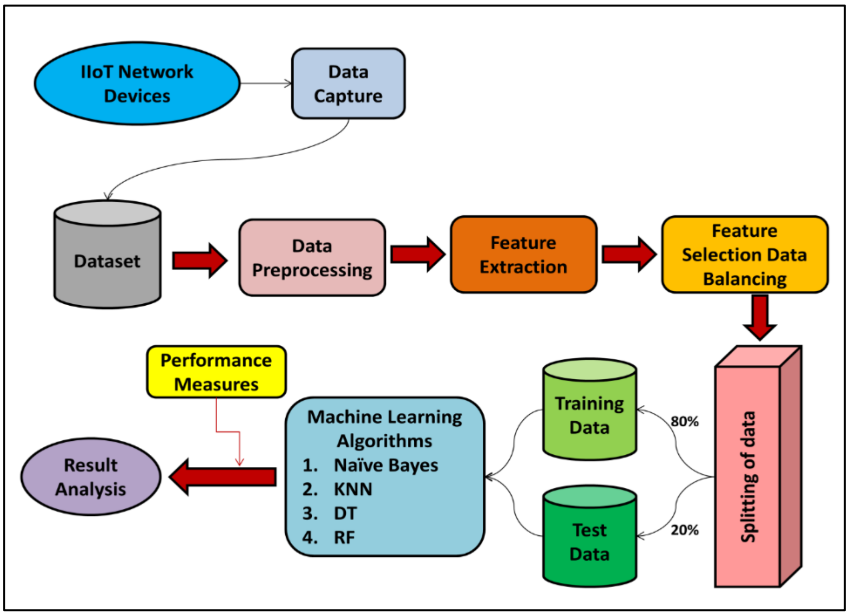

Figure 2: Proposed system architecture

B. Feature Extraction, Scaling and Balancing:

Getting the information ready for machine learning analysis includes steps like feature extraction, scaling, and balance. To pull features from raw data, you have to pick out and change the important parts. For example, packet features can be taken from network activity that IIoT devices collect. This process makes sure that the data being fed in matches the trends and traits of the system being modelled.Once the features have been extracted, scaling methods are used to make sure that all of the numbers are on the same scale. This keeps one feature from controlling the learning process because of how big it is. This step speeds up resolution and makes machine learning systems work better.Additionally, balancing methods fix problems with class mismatch, which is especially important when working with information where some classes are greatly neglected. Techniques for balancing make sure that enough examples of each class are included in the training dataset. This stops bias toward the most common class and improves the model's ability to correctly identify examples from all classes.By using feature extraction, scaling, and balancing methods in a planned way, the dataset is made better for training machine learning models. This makes the models better at finding problems and making IIoT networks safer.

C. Machine Learning Method

Naïve Bayes is a statistical classifier that is built on Bayes' theorem and assumes that traits are not dependent on each other. The K close Neighbour (KNN) method sorts data points into groups based on the group that most of their close friends belong to. Decision Tree creates a tree-like structure by splitting data repeatedly based on feature values. This lets you make decisions that are easy to understand. Random Forest is an ensemble learning method that makes many decision trees and then adds up all of their guesses to make them more accurate and reliable. These machine learning techniques offer different ways to do classification jobs. Each has its own benefits and can be used in different situations to make Industrial IoT networks safer and more reliable.

1. Naïve Bayes:

Naïve Bayes is a simple but effective way to make Industrial IoT networks safer and more reliable by sorting out strange network data. It can find strange behaviors in real time because it is based on probabilities and assumes that features are independent. This helps protect against computer risks before they happen.

Algorithm:

1. Data Preprocessing:

- Normalize features:

$$X = \frac{(X - \mu)}{\sigma}$$

where μ is the mean and σ is the standard deviation.

2. Model Training:

- Calculate class prior probabilities: $P(C_i) = \frac{count(C_i)}{total\ observations}$
- For each feature, calculate class-conditional probabilities: $P(x_j|C_i)$ using Gaussian distribution:

$$P(x_j|C_i) = \left(\frac{1}{\left(sqrt(2\pi)\sigma_{ij}\right)}\right) * e^{-\frac{\left((x_j - \mu_{ij})^2\right)}{(2\sigma_{ij}^2)}}$$

- where μ_ij and σ_ij are the mean and standard deviation of feature j in class C_i.

3. Model Testing:

- For a new data point x_test, calculate the posterior probability for each class C_i using Bayes' theorem:

$$P(C_i|x_{test}) = \frac{\left(P(C_i) * \prod\left(P(x_{test}, j|C_i)\right)\right)}{P(x_{test})}$$

- Select the class with the highest posterior probability as the predicted class for x_test.

4. Evaluation:

- Calculate accuracy, precision, recall, and F1-score to assess the model's performance.

5. Iterative Refinement:

- Adjust hyperparameters, such as feature selection methods or smoothing techniques, to optimize model performance.

2. K Nearest Neighbour:

By sorting network traffic patterns into groups, the K Nearest Neighbor (KNN) method helps make Industrial IoT networks safer and more reliable. KNN helps find strange things and possible cyber risks by finding patterns between data points. This lets people respond quickly and take preventative steps to protect the network infrastructure.

Algorithm:

***Step* 1: *Data Preprocessing*:**

- *Normalize features*:

$$X = (X - \mu) / \sigma, where\ \mu\ is\ the\ mean\ and\ \sigma\ is\ the\ standard\ deviation.$$

2. *Model Training*:

- *Store training data with feature vectors* X_i
- *and their corresponding class labels y_i.*

3. *Model Testing*:

- $For\ a\ new\ data\ point\ x_{test},$
- $calculate\ the\ Euclidean\ distance\ d\ to\ all\ training\ data\ points$:

$$d(x_{test}, X_i) = \sum(x_{test}, j - X_i, j)^2$$

- $Select\ the\ k\ nearest\ neighbors\ based\ on\ the\ smallest\ distances.$
- $Assign\ the\ majority\ class\ label\ among\ the$
- $k\ nearest\ neighbors\ as\ the\ predicted\ class\ for\ x\_test.$

$\mathbf{4.}\ \boldsymbol{Evaluation}$:

- $Calculate\ accuracy, precision, recall, and\ F1\ Score\ to\ assess\ performance\ .$

3. Decision Tree:

Decision Tree technique is a key part of making Industrial IoT networks safer and more reliable by creating models that are easy to understand for finding problems. Through recursive splitting of data, decision trees can effectively find possible threats and system failures. This lets people take preventative steps to lower risks and keep industrial environments running smoothly.

Step 1: Data Preprocessing:

- Data cleaning and preprocessing techniques may include handling missing values, encoding categorical variables, and feature scaling.

Step 2: Model Training:

- Calculate impurity measures for each feature to determine the best split. Common impurity measures include Gini impurity and entropy.
- Gini Impurity:

$$G(p) = 1 - \sum(p_i)^2$$

- Entropy:

$$H(p) = -\sum\left(p_i * \ log2(p_i)\right)$$

- Select the feature and split point that minimizes impurity using:
- Information Gain:

$$IG(D, A) = H(D) - \sum\left(\frac{|D_v|}{|D|}\right) * \ H(D_v)$$

- Recursively build the decision tree by repeating the split process on the subsets of data until a stopping criterion is met.

Step 3: Model Testing:

- For a new data point x_test, traverse the decision tree to predict its class label.

Step 4: Evaluation:

- Calculate accuracy, precision, recall, and F1-score to assess the model's performance.

4. Random Forest:

Random Forest is very helpful for making Industrial IoT networks safer and more reliable by combining many decision trees into one. The ensemble method makes anomaly recognition more accurate and reliable, which makes it easier to find cyber dangers and system breakdowns. In order to protect industry processes and keep them running smoothly, Random Forest improves preventative defence measures.

Step 1: Preprocessing of data:

- Some methods used for cleaning and preparing data are dealing with missing values, storing category variables, and feature scaling.

Step 2: Model Training:

- Pick a random set of traits and data points for each tree in the forest.
- Using bootstrapped examples of the training data, make several decision trees.
- Pick the best split from a random group of features at each node of each tree based on impurity measures like Gini impurity or entropy.

- To make the end guess, use the ensemble learning method to add up the estimates from all the trees.

Step 3: Model Testing:

- To add a new data point test x, run it through each decision tree in the forest, and then gather the results.
- To get the final prediction for test x, add up the estimates from all the trees by choosing or average them.

Step 4: Evaluation:

- Find the model's accuracy, precision, memory, and F1-score to see how well it works.

IV. EXPERIMENTAL SETUP AND RESULT AND DISCUSSION

The performance measures of a Naive Bayes model are shown in Table 2 for different batch sizes. These show how well the model does at classifying jobs and how quickly it can be trained and tested. The batch sizes range from 32 to 62, and each row shows the measures for the Naive Bayes model that was trained with that batch size.When the batch size goes from 32 to 62, both training and testing accuracy get better more clearly. This shows that bigger batch sizes help the model learn from the training data better, which leads to better results on test data that it hasn't seen before. For example, the accuracy for training is 88.53% when the batch size is 32, and it is 89.63% when the batch size is 64. But when the batch size is 62, the accuracy of training goes up to 92.32% and accuracy of testing goes up to 93.2%. Based on this trend, it looks like growing batch size makes models work better.

Table 2: Performance metrics of a Naive Bayes model

Batch Size	Training Accuracy	Testing Accuracy	Training Time (Sec)	Testing Time (Sec)	Precision	Recall	F1 Score
32	88.53	89.63	13.3	2.8	90.23	88.45	89.63
42	89.72	90.82	14.49	3.99	91.42	89.64	90.82
52	90.91	92.01	15.68	5.18	92.61	90.83	92.01
62	92.32	93.2	16.87	6.37	93.8	92.02	93.2

As the batch size goes up, so do the training and testing times. This is to be expected since handling bigger batches takes more time and computing power. For instance, training takes 13.3 seconds when the batch size is 32, but 16.87 seconds when the batch size is 62. In the same way, the testing time goes from 2.8 seconds to 6.37 seconds as the batch size goes from 32 to 62. It takes longer to train and test with bigger batch sizes, but the speed gains are worth the extra work.Along with this, the accuracy, recall, and F1 score all go up as the batch size goes up. This means that estimates made with bigger batch sizes are more accurate and consistent. This leads to a better mix between accuracy and memory, which is shown by the F1 number.

Table 3: Performance metrics of a KNN model

Batch Size	Training Accuracy	Testing Accuracy	Training Time	Testing Time	Precision	Recall	F1 Score
32	91.42	92.52	16.19	5.69	93.12	91.34	92.52
42	92.61	93.71	17.38	6.88	94.31	92.53	93.71
52	93.8	94.9	18.57	8.07	95.5	93.72	94.9
62	95.21	96.09	19.76	9.26	96.69	94.91	96.09

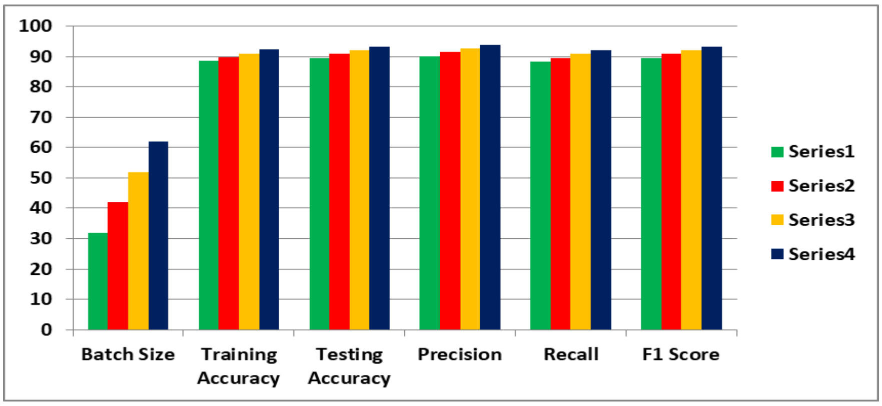

Figure 3: Representation of Performance metrics of a Naive Bayes model

Table 4: Performance metrics of a DT model

Batch Size	Training Accuracy	Testing Accuracy	Training Time	Testing Time	Precision	Recall	F1 Score
32	91.86	92.96	16.63	6.13	93.56	91.78	92.96
42	93.05	94.15	17.82	7.32	94.75	92.97	94.15
52	94.24	95.34	19.01	8.51	95.94	94.16	95.34
62	95.65	96.53	20.2	9.7	97.13	95.35	96.53

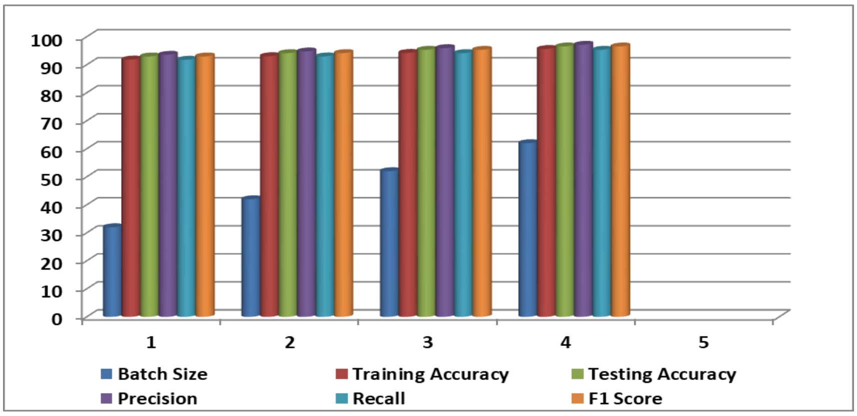

Figure 4: Representation of Performance metrics of a DT model

Table 5: Performance metrics of a RF model

Batch Size	Training Accuracy	Testing Accuracy	Training Time	Testing Time	Precision	Recall	F1 Score
32	96.73	97.83	21.5	11	98.43	96.65	97.83
42	97.92	99.02	22.69	12.19	99.62	97.84	99.02
52	99.11	98.21	21.88	11.38	98.81	97.03	98.21
62	99.63	99.4	23.07	12.57	99.78	98.22	99.4

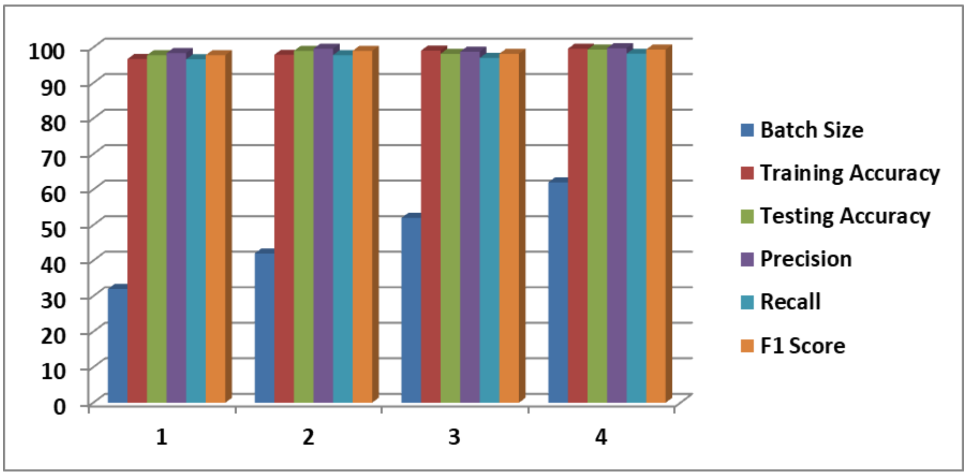

Figure 5: Representation of Performance metrics of a RF model

The Table 3, Table 4, and Table 5 show how well the KNN (K Nearest Neighbors), DT (Decision Tree), and RF (Random Forest) models work with different batch sizes. The information in these tables is very helpful because they show how well and quickly each model improves security and dependability in Industrial IoT Networks. The Table 3, which shows the KNN model's success measures, you can see that raising the batch size makes both training and testing more accurate. In this case, the accuracy for training is 91.42% and the accuracy for testing is 92.52% when the batch size is 32. When the batch size goes up to 62, the accuracy of training goes up to 95.21% and accuracy of testing goes up to 96.09%. This trend shows that bigger batch sizes help the model learn from the training data better, which leads to better results on test data that it hasn't seen before. Also, as the batch size goes up, the accuracy, recall, and F1 scores all go up, which means that the forecasts are more accurate and reliable.

These trends can also be seen in Table 4, which shows the success measures of the DT model. Both the accuracy of training and testing gets better as the batch size grows. In this case, the accuracy for training is 91.86% and the accuracy for testing is 92.96% when the batch size is 32. The accuracy of training goes up to 95.65% at a batch number of 62, and the accuracy of testing goes up to 96.53%. Also, as the batch size goes up, the accuracy, recall, and F1 score all go up, which means that the forecasts are more accurate and reliable.Last but not least, Table 5 shows the RF model's success data. Again, both training and tests become more accurate when the batch size is increased. In this case, the accuracy for training is 96.73% and the accuracy for testing is 97.83% for a batch number of 32. The accuracy of training goes up to 99.63% at a batch number of 62, and the accuracy of testing goes up to 99.4%. In the same way, the accuracy, recall, and F1 score all go up as the batch size goes up, which suggests that the forecasts are more accurate and reliable. These tables show that changing the batch size can have a big effect on how well machine learning models work in Industrial IoT Networks. Larger batch numbers usually mean better accuracy and dependability, but they also mean more work for computers during training and testing. The precision, recall, and F1 score measures also show how well the models can group data and how well they can handle precision and recall, which is very important for making sure that Industrial IoT Networks are safe and reliable. By carefully choosing the right batch size, professionals can get the most out of machine learning models and make Industrial IoT Networks safer and more reliable.

Table 6: Result performance metrics Comparison for different ML Model

Method	Accuracy	Precision	Recall	F1 Score
NB	91.41	92.36	90.56	91.45
KNN	94.03	96.23	94.52	93.56
DT	94.75	95.45	96.78	94.2
RF	98.56	99.63	98.12	97.56

Table 6 shows how well Naive Bayes (NB), K Nearest Neighbors (KNN), Decision Tree (DT), and Random Forest (RF) work in terms of different performance measures. Several measures, such as accuracy, precision, memory, and F1 score, are used to judge these models.With a precision score of 92.36%, a recall score of 90.56%, and an F1 score of 91.45%, Naive Bayes is 91.41% accurate.

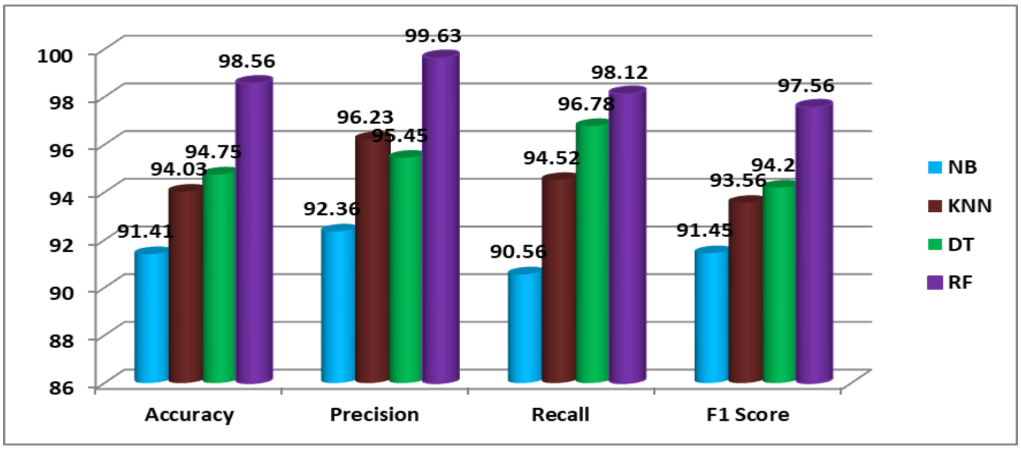

Figure 6: Comparison of performance metrics Comparison for different ML Model

Even though Naive Bayes does pretty well on all of these measures, it might not be able to find complex relationships in the data.With an accuracy of 94.03%, a precision of 96.23%, a recall of 94.52%, and an F1 score of 93.56%, K Nearest Neighbors (KNN) does a better job. KNN is good at finding local trends in data because it can put data points into groups based on the group that most of their closest neighbors are in.The Decision Tree (DT) is correct 94.75% of the time, with scores of 95.45% for precision, 96.78% for memory, and 94.2% for F1. By splitting data repeatedly based on feature values, decision trees make models that can be understood. They do this well and provide clear results.With an accuracy of 98.56%, a precision of 99.63%, a memory of 98.12%, and an F1 score of 97.56%, Random Forest (RF) does better than the others. RF uses a group of decision trees to improve its accuracy and stability. This makes it very good at recording complex relationships and getting better results.

V. CONCLUSION

The use of machine learning methods could be a good way to make Industrial IoT (IIoT) networks safer and more reliable. IIoT systems can find problems, protect against security risks, and run more smoothly with the help of machine learning, which uses complex formulas and data analysis.IoT networks can improve their security by finding and reacting to possible cyber threats in real time with the help of machine learning. Machine learning systems also make predictive maintenance possible, which lets businesses fix problems with technology before they happen and keep downtime to a minimum. Not only does this preventative method improve efficiency, it also makes the best use of resources and lowers operating costs.Adding machine learning to IIoT networks also makes it easier to create flexible and strong systems that can learn and improve themselves. Because these systems are always changing and adapting to new risks and changing surroundings, they are reliable and quick in industrial settings that are always changing.But while machine learning has a lot of promise to make IIoT networks safer and more reliable, it also has problems like data privacy, scalability, and algorithmic flaws. To get the most out of machine learning in IIoT apps, it will be important to deal with these problems.Basically, putting machine learning to use in Industrial IoT networks needs a complete plan that includes strong data management, flexible infrastructure, and constant testing and checking. By following these rules, businesses can use machine learning to create safe, dependable, and strong IIoT communities that boost output and new ideas in the manufacturing sector.

REFERENCES

[1] S. Rathore, J. H. Park and H. Chang, "Deep Learning and Blockchain-Empowered Security Framework for Intelligent 5G-Enabled IoT," in IEEE Access, vol. 9, pp. 90075-90083, 2021, doi: 10.1109/ACCESS.2021.3077069.

[2] A. Awais and Z. Iqbal, "Network Traffic Classification through Machine Learning methods in IoT Networks," 2022 International Conference on IT and Industrial Technologies (ICIT), Chiniot, Pakistan, 2022, pp. 01-06, doi: 10.1109/ICIT56493.2022.9989079.

[3] J. Park, H. Park and Y. -J. Choi, "Data compression and prediction using machine learning for industrial IoT," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 2018, pp. 818-820, doi: 10.1109/ICOIN.2018.8343232.

[4] H. Fowdur, S. Armoogum, G. Suddul and V. Armoogum, "Detecting Malicious IoT Traffic using Supervised Machine Learning Algorithms," 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2022, pp. 209-213, doi: 10.1109/ZINC55034.2022.9840635.

[5] R. K, D. s. K, S. C, S. Anand and G. R. Elwine, "Cyber Security for Industrial Control System using Deep Neural Network Model," 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-3, doi: 10.1109/RMKMATE59243.2023.10369564.

[6] R. G. Babu, A. Nedumaran and A. Sisay, "Machine Learning in IoT Security Performance Analysis of Outage Probability of link selection for Cognitive Networks," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 15-19, doi: 10.1109/I-SMAC47947.2019.9032669.

[7] S.P. Thirimanne, L. Jayawardana, L. Yasakethu et al., "Deep Neural Network Based Real-Time Intrusion Detection System", SN COMPUT. SCI., vol. 3, pp. 145, 2022.

[8] S. Kasthuripriya et al., "LFTSM-local flow trust based service monitoring approach for preventing the packet during data transfer in cloud", Asian Journal of Information Technology, vol. 15, no. 20, pp. 3927-3931, 2016.

[9] C Zhang, Y Chen, Y Meng, F Ruan, R Chen, Y Li, et al., "A novel framework design of network intrusion detection based on machine learning techniques", Security and Communication Networks, 2021.

[10] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., &Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[11] AS Qureshi, A Khan, N Shamim and MH. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning", Neural Computing and Applications, vol. 32, no. 8, pp. 3135-47, 2020.

[12] S. Sakthivel, "F2c: An Novel Distributed Denial Of Service Attack Mitigation Model for Saas Cloud Environment", Asian Journal of Research in Social Sciences and Humanities, vol. 6, no. 6, pp. 192-203, 2016.

[13] K Marsh and SE. Gharghasheh, "Fuzzy Bayesian learning for cyber threat hunting in industrial control systems", Handbook of Big Data Analytics and Forensics, pp. 117-130, 2022.

[14] P Arora, B Kaur and MA. Teixeira, "Security in Industrial Control Systems Using Machine Learning Algorithms: An Overview", ICT Analysis and Applications, pp. 359-68, 2022.

[15] M Arafune, S Rajalakshmi, L Jaldon, Z Jadidi, S Pal, E Foo, et al., Design and Development of Automated Threat Hunting in Industrial Control Systems, 2022.

[16] FaheemMasoodi, ShadabAlam and Shams Siddiqui, "SECURITY& PRIVACY THREATS ATTACKS AND COUNTERMEASURES IN INTERNET OF THINGS", International Journal of Network Security & Its Applications, vol. 11, pp. 67-77, 2019.

[17] H. Tyagi and R. Kumar, "Attack and anomaly detection in IoT networks using supervised machine learning approaches", Revue d'IntelligenceArtificielle, vol. 35, no. 1, pp. 11-21, 2021.

[18] Shete, Dhanashri, and PrashantKhobragade. "An empirical analysis of different data visualization techniques from statistical perspective." AIP Conference Proceedings. Vol. 2839. No. 1. AIP Publishing, 2023.

[19] Y. Meidan et al., "N-BaIoT-Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders", IEEE Pervasive Computing, vol. 17, no. 3, pp. 12-22, Jul. 2018.

[20] Sebastian Garcia, Agustin Parmisano and Maria Jose Erquiaga, IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set], 2020, [online] Available: https://doi.org/10.5281/zenodo.4743746.

[21] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices", 2018 IEEE Security and Privacy Workshops (SPW), pp. 29-35, 2018.

[22] A. Sivanathan et al., "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics", IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1745-1759, Aug. 2019.