iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60450

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Network Vulnerability Analysis of the Industrial Internet of Things Using Machine Learning

Mansi Samir Pathare[1], Divakar Mallah[2], Asthma Bhagat[3], Prof. Md. Ameen[4]
Computer Science & Engineering(AIML)Department, Mumbai University

Abstract: ***Over this period of time, there has been a significant increase in the quantum of data reused over the Internet due to the gradual increase in technology operation. The massive volume of data being transferred over the Internet raises the need for data security, which is where vulnerability discovery systems (IDS) come into play and aid in the discovery of any pitfalls to virtual security. An intrusion discovery system, or IDS, is a device that keeps an eye on and evaluates data in order to find any cases of network or system intrusion. Hackers use colorful ways to gain access to a network. In order to classify attacks, describe them whenever an attack occurs, and determine which machine literacy algorithm is most applicable for relating the attack, the proposed Vulnerability discovery system is being enforced using slice- edge technologies, similar to machine literacy algorithms.***
Keywords: Industrial Internet of effects, Machine literacy, Intrusion, Intrusion Discovery System, Denial of Service, star element Analysis, Support Vector Machine, Random Forest, Decision Tree, KNN Algorithm, Logistic Regression, cautions, False Cons, False Negatives.

I. INTRODUCTION

This document An Vulnerability detection system is an apparatus or software program designed to keep an eye out for malicious activity or system or network violations. A Vulnerability system mixes outputs from various sources and separates malicious activity from false alarms using alarm filtering techniques. While monitoring networks for potentially malicious activity, intrusion detection systems are also prone to false alarms. Therefore, when deploying products for the first time, organizations must fine-tune them. In order to distinguish between malicious activity and regular network traffic, intrusion detection systems must be configured correctly. well as the collection, analysis, and logging of data. Making use of Internet of Things technology in industrial control systems is the main idea behind the Industrial Internet of Things (IIoT). Industrial control systems, or ICSs, have long been used to monitor industrial machinery and processes. They are a vital component of critical infrastructures. In addition to logging all events that occur in the industrial systems, they also conduct real-time data collection and analysis, device interaction, and monitoring. The optimization and automation of industrial processes are made possible by the use of IoT technology in these systems, which improves network intelligence and security.

A. Vulnerability analysis Architecture:

Using machine learning algorithms like decision trees, regression, random forests, and KNN, we have gathered the dataset for the intrusion detection system, which includes the following details from the KDD dataset. Details of the Data Set. The process of gathering data includes choosing high-quality data to be analyzed. To implement machine learning in this case, we used the KDD vulnerability dataset that was obtained from uci.edu. Finding approaches and resources for gathering thorough and pertinent data, analyzing it using statistical methods, and evaluating the findings are the duties of a data analyst. Data visualization: A lot of information is simpler to comprehend and evaluate when it is presented graphically.

1) *Data Collection:* The process of collecting data includes choosing high-quality data to be analyzed. For the machine learning implementation, we used the KDD intrusion dataset, which was obtained from uci.edu. Finding methods and resources for gathering thorough and pertinent data, analyzing it using statistical methods, and reporting findings are all part of a data analyst's job description.
2) *Visualization of Data*: A lot of information is simpler to comprehend and evaluate when it is presented graphically. Certain companies mandate that a data analyst be proficient in creating charts, diagrams, slideshows, and templates. The data visualization component of our method displays the intrusion detection rates.

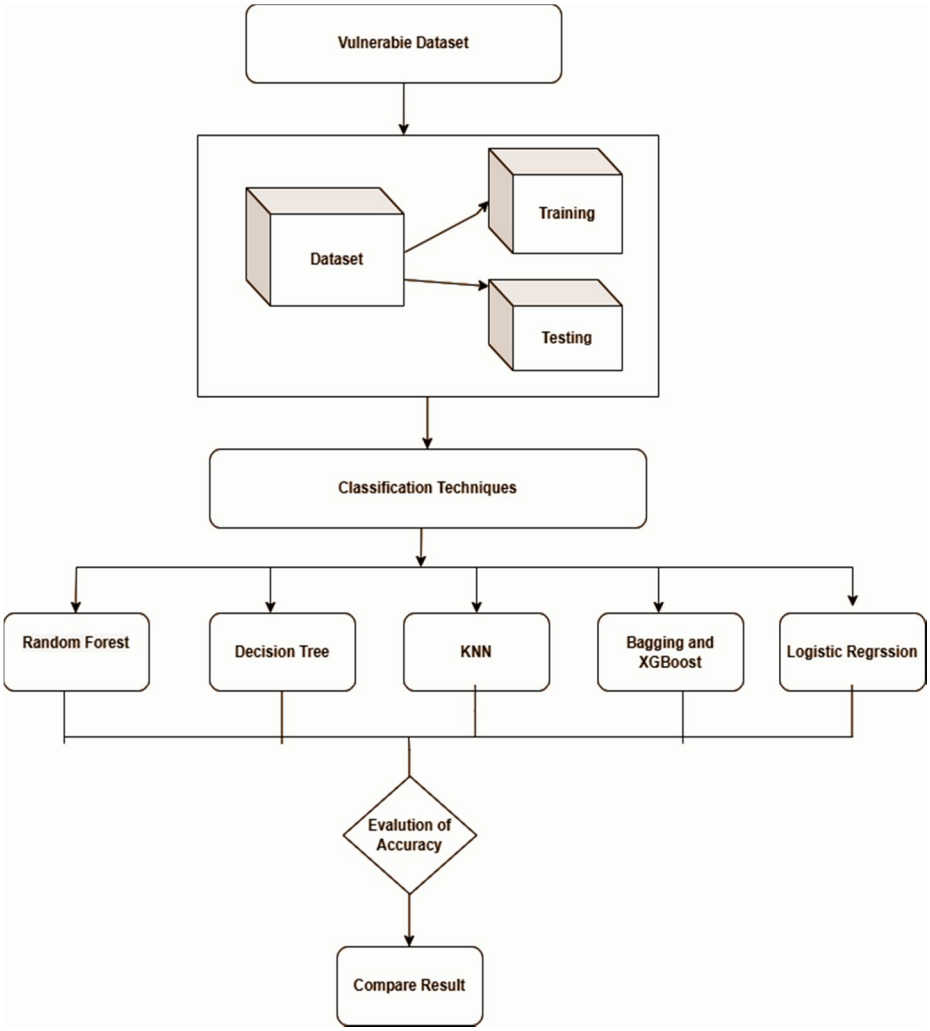


Fig 1.1: Vulnerability Architecture proposed

3) *Dataset Splitting*: The dataset can be divided into two parts. Where part one is training dataset and testing dataset. The training dataset can be divided into 70%to 80%. The testing dataset can be divided into 20% to 30%. Training, test, and validation sets are the three subsets into which a dataset used for machine learning should be divided. Training set: When training a model and defining its ideal parameters, a data scientist uses a training set. The model learns from it. Test set: To assess the trained model's capacity for generalization, a test set is necessary. After training over training data, the latter refers to a model's capacity to spot patterns in fresh, untainted data. To prevent overfitting of the model—the previously mentioned inability to generalize—it is imperative to utilize distinct subsets for training and testing.
4) *Model Training:* Model training can start once a data scientist has preprocessed the gathered data and divided it into train and test sets. The algorithm must be "fed" training data during this procedure. When you use predictive analysis to get an answer, an algorithm will process data and produce a model that can identify a target value (attribute) in fresh data. Model development is the goal of model training.
5) *Model Testing and Evaluation:* This step aims to create the simplest model that can generate a target value quickly and accurately enough. Model tuning is one way a data scientist can accomplish this. It is the process of fine-tuning model parameters to maximize algorithmic performance. Not only does the test data contain specific attack types that are not present in the training data, but it also differs in probability distribution from the training data. This adds realism to the task. Some intrusion experts contend that the majority of new attacks are really variations of well-known ones, and that new variants can often be identified by their "signature.". There are 24 attack types in total across the datasets for training, and an additional 14 types are present solely in the test data.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

B. Implementation of Machine learning algorithm:

There are various machine learning algorithms which can be useful for vulnerability detection and analysis which are: Decision Tree, Logistic Regression, KNN, and Random Forest are four of the machine learning algorithms that are taken into consideration for intrusion detection. For the expected accuracy and error values, the predicted value is compared.

1) Logistic Regression Algorithm:

The Logistic Regression Algorithm is a powerful and simple predictive model analysis technique used in machine learning. It is typically used for binary classification problems, predicting the probability of a binary outcome using a logit function. Around 60% of classification problems can be solved using this algorithm. It is a special case of linear regression, predicting probabilities of outcomes using a log function. In simple terms, it predicts scores on one variable based on scores from a second variable, with the predicted variable called the Criterion Variable.

a) Sigmoid Function: The sigmoid function is a mathematical function that has the ability to map any real number between o and 1, giving it the shape of a letter "S." A mathematical function with a distinctive "S"-shaped curve, or sigmoid curve, is called a sigmoid function. Any value in the domain is converted to a number between 0 and 1.

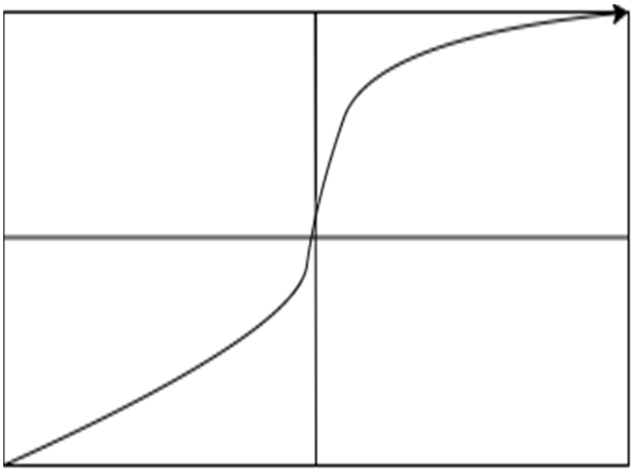


Fig.1.2: Sigmoid Curve

Any mathematical function with a distinctive S-shaped or sigmoid curve on its graph is called a sigmoid function. Logistic regression have sigmoidal function which contain s-shape curve that can we use for regression

2) Decision Tree Algorithm

One kind of supervised learning algorithm that is commonly used in classification issues is the decision tree. It functions with both continuous and categorical input and output variables. Using the most significant splitter or differentiator in the input variables, we divide the sample into two or more homogeneous sets (or sub-populations) in this technique. An internal node in a decision tree indicates an attribute test, a branch shows the result, and a leaf indicates the choice made after computing the attribute.

The main goal of using decision trees is to build a training model that, by learning decision rules deduced from previous data (training data), can be used to predict a target variable's class or value. Compared to other classification algorithms, the Decision Tree algorithm is very simple to understand. The Decision Tree algorithm uses tree representation to attempt to solve the problem. Every leaf node in the tree corresponds to a class label, and every internal node to an attribute.

3) Random Forest Algorithm

With n cases in the training dataset, the Random Forest Algorithm is used. N subsamples with replacement are randomly selected from these n cases. Each tree is constructed using these randomly selected subsamples from the training dataset. Choose a number m such that m < k, given that there are k variables for input. Every node has k variables, from which m are chosen at random. The split of the node is determined by selecting the split that maximizes these m variables. As the forest expands, the value of m remains unaltered. Without any pruning, every tree is allowed to reach its full potential.

4) K Nearest Neighbor (KNN) Algorithm

The KNN algorithm calculates the separation between a set of scenarios in the data set and a query scenario. Using a distance function d(x,y), where x,y are scenarios made up of N features and x={x1,…,xN} and y={y1,…,yN}.

The type of data affects the similarity measure. The Euclidean distance can be applied to real-valued data. Other data types, like binary or categorical data, can also be utilized, as well as Hamming distance. In order to model the issue and produce predictive decisions, instance-based algorithms use data instances (or rows). Since every training observation is kept as a component of the model, the KNN algorithm represents an extreme version of instance-based techniques.

II. LITERATURE SURVEY

1) In this system that uses either an anomaly-based approach or a signature-based approach. These machine learning algorithms—Decision Tree, Random Forest, Logistic Regression, and KNN—will assist in mitigating the drawbacks of the current intrusion detection systems. Of the three, Random Forest is the most effective for the following reasons: 1. Excellent Precision, 2. Employs Regression 3 and Classification Techniques Together. 4. It applies the group method, or bagging.Keywords: Decision Tree, Random Forest, machine learning ,Methods used: Because the signature-based detection method can only identify known attacks, it can occasionally produce false positive results. On the other hand, the anomaly-based detection method produces a higher false positive rate by raising alarms for even valid events and network traffic.

2) In this paper methods, resources, and obstacles. The methods under consideration in this work are: Signature-based intrusion detection systems (SIDS) Anomaly-based intrusion detection systems (AIDS) Knowledge-based machine learning. The primary benefit of AIDS is its capacity to detect zero-day attacks since it does not depend on a signature database to identify abnormal user behavior. Keywords: Signature- based intrusion detection, machine learning ,Methods used: The primary benefit of AIDS is its capacity to detect zero-day attacks, as it eliminates the need for a signature database in order to identify unusual user behaviour.

3) In today's environment of data communication, network and system security is critical. Unauthorized intrusion by hackers and intruders can result in numerous successful attempts to bring down networks and web services. As secured systems evolve, new threats and related countermeasures to these threats also emerge. Among these are Intrusion Detection Systems (IDS). An intrusion detection system's primary job is to keep resources safe from harm. It examines and forecasts user behavior, determining whether a given behaviour is deemed typical or an attack. To find network intrusions, we employ Support Vector Machine (SVM) and Rough Set Theory (RST). After capturing packets from the network, the data is pre-processed and its dimensions are decreased using RST. The RST-selected features will be sent to the SVM model for testing and learning, respectively. Since AIDS does not rely on a signature database to identify abnormal user activity, its primary benefit is its ability to detect zero-day attacks.Keywords: Intrusion Detection System, Remote System Threat, Support Vector Machine, PCA ,Methods used: In order to lower the number of features from 41 to 29, they suggested an intrusion detection technique based on an SVM system on an RST. We also contrasted PCA's performance with that of Rough Set Theory (RST).

4) Using map reduce operations to detect web-based DDoS attacks in a cloud computing environment. The most dangerous element when it comes to network security risks in the cloud computing environment is distributed denial of service attacks. In order to quickly detect attacks in cloud computing environments, this study suggests integrating MapReduce processing with HTTP GET flooding among DDOS attacks. By using HTTP GET flooding, this technique can guarantee that the target system is available for precise and trustworthy detection. In tests, the processing time for performance assessment contrasts the Snort detection with a pattern recognition of attack features. Based on processing time, the suggested method outperforms the Snort detection method in the experiment results.Keywords: MapReduce processing, HTTP GET flooding, DDoS attacks, web security, Methods used: Suggested integrating MapReduce processing with HTTP GET flooding to detect attacks quickly in cloud computing environments. By using HTTP GET flooding, this technique can guarantee that the target system is available for precise and trustworthy detection.

III. DESIGN

A. *UML*

A general-purpose modelling language is called Unified Modelling Language (UML). UML's primary goal is to establish a common framework for visualizing a system's design process. It resembles blueprints from other engineering specialties quite a bit. UML diagrams are used to show a system's structure and behaviour. UML aids in the modelling, design, and analysis processes for system architects, software engineers, and businesspeople. Unified Modelling Language was standardized by the Object Management Group (OMG) in 1997.

1) *Use Case Diagram:* Using actors and use cases, a use case diagram captures the requirements and functionality of the system. Use cases serve as a model for the functions, duties, and services that a system must provide. Use cases show high-level features and the way a user will interact with the system. The fundamental ideas of Unified Modelling language modelling are use-cases.
2) *Class Diagram:* Using design elements like classes, packages, and objects, class diagrams represent the contents and structure of a class. Three perspectives are described in a class diagram when designing a system: conceptual, specification, and implementation. Three elements make up a class: name, attributes, and operations. Class diagrams also show relationships like inheritance, associations, and containment. The most prevalent type of relationship in a class diagram is the association relationship.

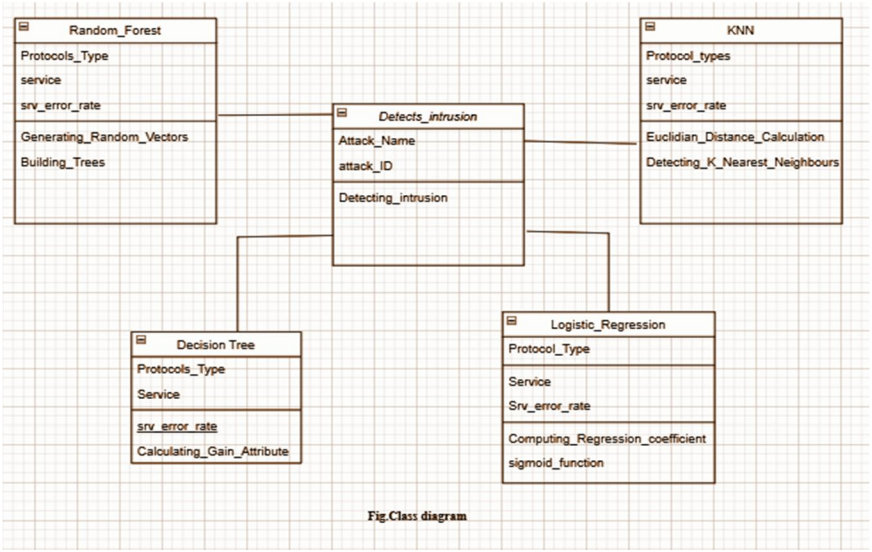


Fig.3.1 Class Diagram

3) *Sequence Diagram:* Sequence diagrams show the objects' relative positions in time during an interaction. Time is the vertical dimension, and various objects are the horizontal dimension.

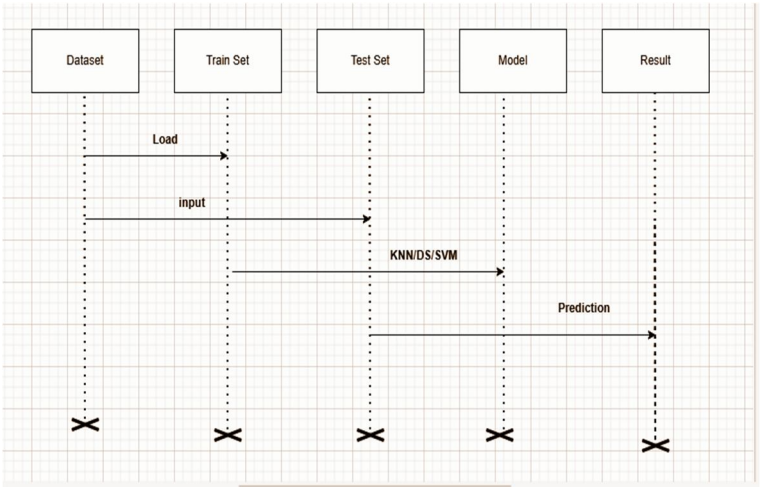


Fig.3.2 Sequence Diagram

Object: An object is defined as an entity with a specific value at a given moment in time, as well as a holder of identity whose values change over time. Actor: An Actor is a cohesive group of roles that system users assume when interacting with the system's use cases. Message: A message is sent when an object sends a signal to another object that receives it.

4) Activity Diagram:

An activity diagram illustrates how one activity flows into another. Within a state machine, an activity is a continuous, non-atomic execution. An activity produces an outcome, such as an action, a state change, or the return of a value. Common components of an activity diagram are action states and activity states. Changes in Direction. Nodes and constraints. s may be present in objects. Activity states and action states are executable atomic computations that are non-decomposable. Activity states are non-atomic, decomposable, and require a certain amount of execution time.

a) Transition: A simple directed line that guides an object from one state to the next. Branching: An open diamond indicates branching, which occurs when there is a different route available. It features one outgoing transition and two or more incoming transitions. Forking and Joining: A fork is the synchronization bar formed when a single flow splits into two or more streams. The term "join" refers to a synchronization bar where two or more flows are combined.

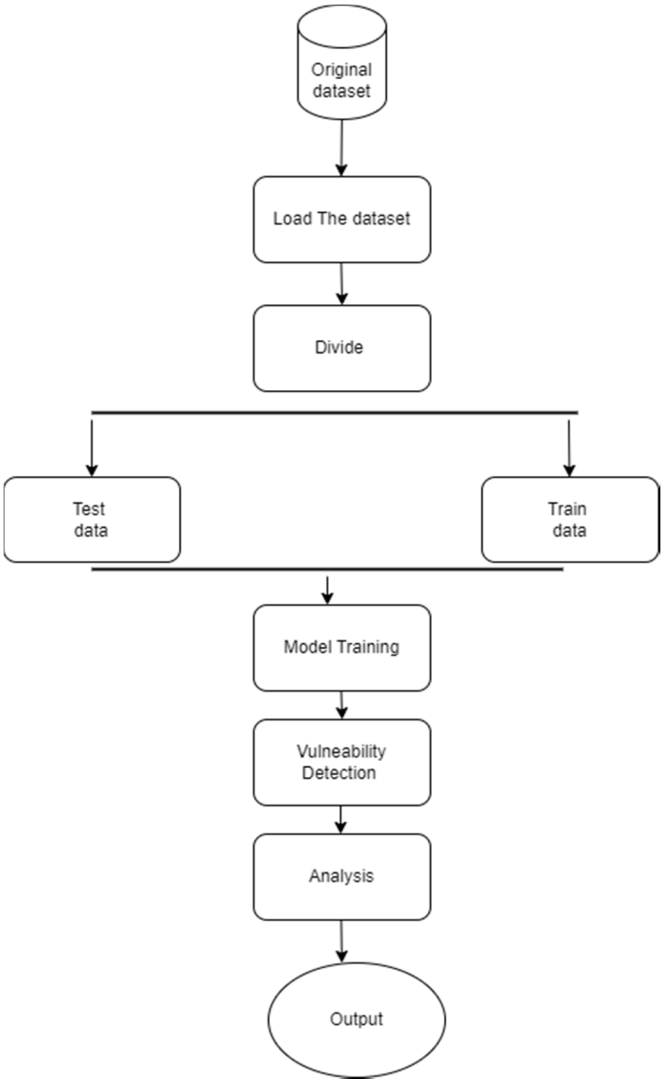


Fig 3.3 Activity Diagram

Swim Lanes: Swim lanes refer to the group work flow. Solid vertical lines divide up each group. Each swim lane is named differently and designates the area of activity. One implements every swim lane.

5) State Chart Diagram: An illustration of a state machine that depicts class behaviour is called a state chart diagram. It represents the dynamic behaviour of objects over time by simulating the lifecycle of objects of each class and displays the actual state changes rather than the processes or commands that cause those changes. It explains how an object transitions between different states.

a) State: The State Chart Diagram comprises several components, the primary ones being the Initial state and Final state. The State represents a state in an object's lifecycle in which it either performs an activity, waits for an event, or satisfies a condition.

b) Transition: This is the relationship between two states that shows an object in the first state moves into the next state after carrying out a certain action.

c) Event: An event is a specific, location-based, significant occurrence in space and time.

6) Deployment Diagram: The Topology of the physical component of a system, where the software components are deployed, is visualized using a deployment diagram. Nodes and their connections make up a deployment diagram. A deployment diagram is used to illustrate the hardware deployment process. The system engineers can use these. Deployment diagrams must be effective because they govern performance, maintainability, and portability.

IV. RESULTS

A. Dataset

	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	...	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std
0	3	2	0	12	0	6	6	6.0	0.00000	0	...	20	0.0	0.0	0	0	0.0	0.0
1	109	1	1	6	6	6	6	6.0	0.00000	6	...	20	0.0	0.0	0	0	0.0	0.0
2	52	1	1	6	6	6	6	6.0	0.00000	6	...	20	0.0	0.0	0	0	0.0	0.0
3	34	1	1	6	6	6	6	6.0	0.00000	6	...	20	0.0	0.0	0	0	0.0	0.0
4	3	2	0	12	0	6	6	6.0	0.00000	0	...	20	0.0	0.0	0	0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
30738	32215	4	2	112	152	28	28	28.0	0.00000	76	...	20	0.0	0.0	0	0	0.0	0.0
30739	324	2	2	84	362	42	42	42.0	0.00000	181	...	20	0.0	0.0	0	0	0.0	0.0
30740	82	2	1	31	6	31	0	15.5	21.92031	6	...	32	0.0	0.0	0	0	0.0	0.0
30741	1048635	6	2	192	256	32	32	32.0	0.00000	128	...	20	0.0	0.0	0	0	0.0	0.0
30742	94939	4	2	188	226	47	47	47.0	0.00000	113	...	20	0.0	0.0	0	0	0.0	0.0

Fig 4.1 Dataset

B. Data Split Set and Training Set

```
0    18184
3    15228
5     6357
2     2213
6     1744
1     1573
4       29
dtype: int64
```

Fig 4.2 Data Split set and training set

C. Machine Learning Model Training

```
Accuracy of DT: 0.9960292949792641
Precision of DT: 0.9960126519428796
Recall of DT: 0.9960292949792641
F1-score of DT: 0.9960148981765187
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      4547
           1       0.99      0.98      0.98       393
           2       0.99      1.00      1.00       554
           3       1.00      1.00      1.00      3807
           4       0.83      0.71      0.77         7
           5       1.00      1.00      1.00      1589
           6       0.99      0.99      0.99       436

    accuracy                           1.00     11333
   macro avg       0.97      0.95      0.96     11333
weighted avg       1.00      1.00      1.00     11333
```

Fig 4.3 Machine learning model training

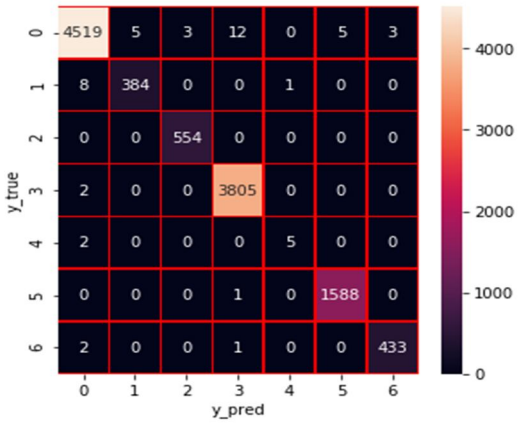


Fig 4.4 Output 1

	DecisionTree	ExtraTrees	RandomForest	XgBoost
0	5	5	5	5
1	3	3	3	3
2	5	5	5	5
3	3	3	3	3
4	2	2	2	2

Fig 4.5 Output 2

```
Accuracy of ET: 0.9920585899585282
Precision of ET: 0.9920197520868573
Recall of ET: 0.9920585899585282
F1-score of ET: 0.9920217765183937
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      4547
           1       0.96      0.97      0.97       393
           2       0.99      1.00      0.99       554
           3       0.99      1.00      1.00      3807
           4       0.80      0.57      0.67         7
           5       1.00      1.00      1.00      1589
           6       0.98      0.97      0.98       436

    accuracy                           0.99     11333
   macro avg       0.96      0.93      0.94     11333
weighted avg       0.99      0.99      0.99     11333
```

Fig 4.6 Output 3

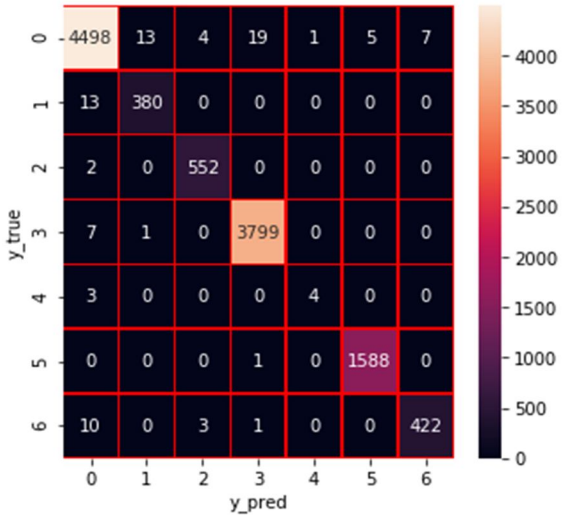


Fig 4.7 Output 4

```
Accuracy of ET: 0.9920585899585282
Precision of ET: 0.9920197520868573
Recall of ET: 0.9920585899585282
F1-score of ET: 0.9920217765183937
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      4547
           1       0.96      0.97      0.97       393
           2       0.99      1.00      0.99       554
           3       0.99      1.00      1.00      3807
           4       0.80      0.57      0.67         7
           5       1.00      1.00      1.00      1589
           6       0.98      0.97      0.98       436

    accuracy                           0.99     11333
   macro avg       0.96      0.93      0.94     11333
weighted avg       0.99      0.99      0.99     11333
```

Fig 4.8 Output 5

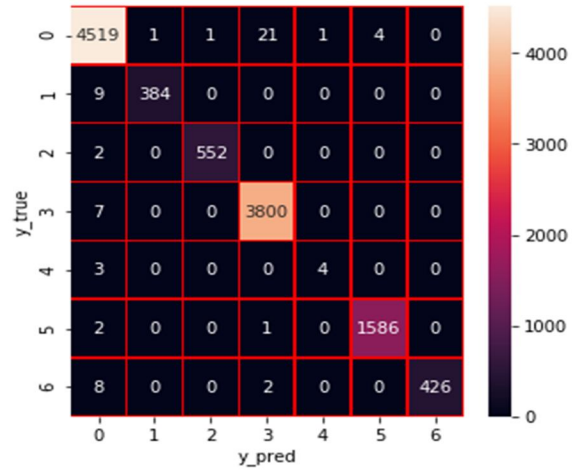


Fig 4.9 Output 6

	DecisionTree	ExtraTrees	RandomForest	XgBoost
0	5	5	5	5
1	3	3	3	3
2	5	5	5	5
3	3	3	3	3
4	2	2	2	2

Fig 4.10 Output 7

```
Accuracy of RF: 0.9924115415159269
Precision of RF: 0.9924641723210192
Recall of RF: 0.9924115415159269
F1-score of RF: 0.9923720836357383
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      4547
           1       0.96      0.97      0.96       393
           2       0.97      1.00      0.98       554
           3       1.00      1.00      1.00      3807
           4       0.83      0.71      0.77         7
           5       1.00      1.00      1.00      1589
           6       1.00      0.93      0.96       436

    accuracy                           0.99     11333
   macro avg       0.96      0.94      0.95     11333
weighted avg       0.99      0.99      0.99     11333
```

Fig 4.11 Output 8

V. CONCLUSION

For IIoT devices, cyber-security is essential. Focusing on the industrial side of IoT technology is essential since there is still a significant gap in providing adequate security for these systems. Security in IT systems has been ensured by the widespread use of big data analytics and machine learning solutions. But the prevalent cyber-risks associated with traditional IT systems differ because of their fundamentally different priorities. Therefore, security for IIoT requires extra care. We have shown how effective machine learning is for enhancing the security of these systems through our discussions and experimental evaluation. The first thing we did in this paper was examine the security vulnerabilities of the four most widely used IIoT protocols. After that, we evaluated the risks associated with the most significant and common vulnerabilities of the IIoT systems and the ways in which machine learning-based remedies could be effective in addressing them. Next, to highlight the areas where security is still required, a review of the literature was done on the machine learning-based anomaly detection techniques.

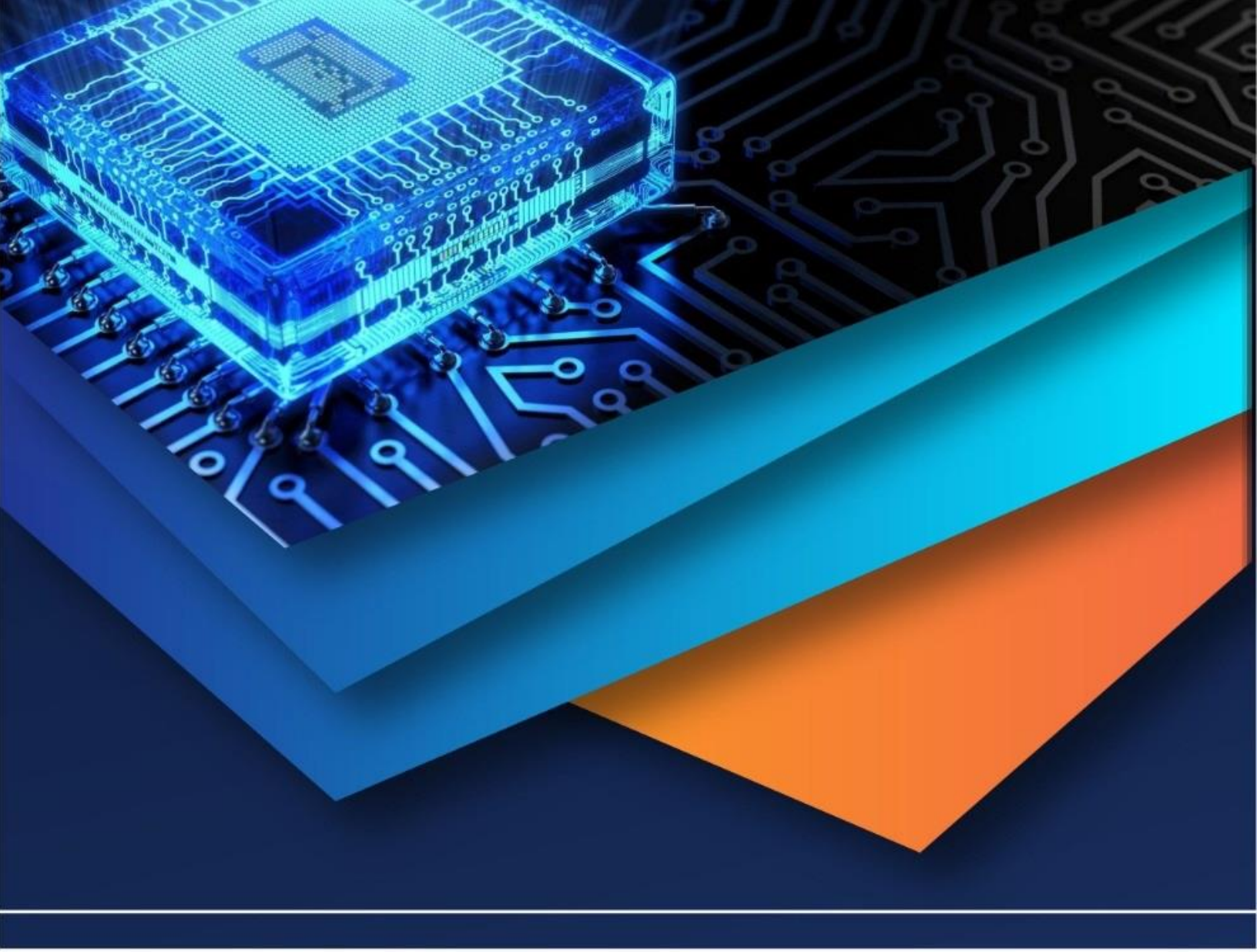
The primary goal of an Intrusion Detection System is to identify and prevent attacks and malicious behavior within a network while minimizing false alarms. Utilizing machine learning algorithms enhances the accuracy and reliability of the IDS output. It also assesses the effectiveness of different machine learning algorithms in detecting attacks. The growing reliance on technology has resulted in vast amounts of data that must be securely processed and stored, emphasizing the importance of security for users.

VI. ACKNOWLEDGMENT

WeThank all those people who supported me and my group members for completion of our project. I thank my guide Dr. Md. Ameen for guiding me throughout. I also want to express my gratitude towards Prof. K.N. Attarde ,head of the Department For his motivation.I want to thank my family for continuously motivating & Supporting me for my research work.

REFERENCES

[1] A B. Athira, V. "Standardization and Classification of Alerts Generated by Intrusion Detection Systems," Pathari, vol. 5, issue 2, 2016,
[2] International Journal on Cybernetics and Informatics. "Intrusion Detection Systems with Correlation Capabilities" by Daniel Johansson and Par Andersson; Yasm Curt, "Prelude as a Hybrid IDS Framework"; March 2009.
[3] Kumar Vinod and Sangwan Prakash Om, "Signature Based Intrusion Detection System Using SNORT"; International Journal of Computer Applications & Information Technology, Vol. I, November 2012,
[4] "An approach for Anomaly based Intrusion detection system using SNORT," Singh Deepak Kumar, Gupta Jitendra Kumar, International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September 2013.
[5] "Intrusion Detection System - A Study," S. Vijayarani, and Maria Sylviaa S., International Journal of Security, Privacy, and Trust Management, Vol. 4, Issue 1, pp. Feb. 31, 2015–44. [6]"Research of Intrusion Detection System Based on Vulnerability Scanner," ICACC, Advanced Computer Control, Yang Guangming, Chen Dongming, Xu Jian, and Zhu Zhiliang, March 2010.
[6] Chakraborty Nilotpal, "Intrusion Prevention and Detection Systems,". P. Porambage, G. Gür, D. P. M. Osorio, M. Liyanage, and M. Ylianttila, “6G security challenges and potential solutions,” in Proc. IEEE Joint Eur. Conf. Netw. Commun. (EuCNC) 6G Summit, 2021, pp. 1–6.
[7] C. de Alwis et al., “Survey on 6G frontiers: Trends, applications, requirements, technologies and future research,” IEEE Open J. Commun. Soc., vol. 2, pp. 836–886, 2021.
[8] X. You et al., “Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts,” Sci. China Inf. Sci., vol. 64, no. 1, pp. 1–74, 2021.
[9] Arshad, J.; Azad, M.A.; Amad, R.; Salah, K.; Alazab, M.; Iqbal, R. A review of performance, energy and privacy of intrusion detection systems for IoT. Electronics 2020, 9, 629. [Google Scholar] [CrossRef]
[10] Mercer, D. Smart Home Will Drive Internet of Things To 50 Billion Devices. Available online 2023
[11] Gavin Wright, A.S.G. What Is a Side-Channel Attack? 2021.Available online on :https://www.techtarget.com/searchsecurity/ definition/side-channel-attack (accessed on 10 March 2023).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY