International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.3 (2024) pp. 341-349
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Performance Evaluation of Predicting IoT Malicious Nodes Using Machine Learning Classification Algorithms

Poorana Senthilkumar S[1*], Wilfred Blessing N. R.[2], Rajesh Kanna R[3] and Karthik S[4]

[1]Dr. N.G.P. Arts and Science College, Coimbatore, Department of Computer Applications, 641048, Tamil Nadu - India
* **Corresponding Author Email:** pooranasenthilkumar@drngpasc.ac.in - **ORCID:** 0000-0001-5731-2337

[2]University of Tech. and Applied Sciences, College of Computing and Information Sciences, Ibri- Sultanate of Oman
Email: wilfred.blessing@utas.edu.om - **ORCID:** 0000-0003-0458-8641

[3]Christ University, Department of Computer Science, 560029, Karnataka - India
Email: rajeshkanna.r@christuniversity.in - **ORCID:** 0000-0001-7228-5031

[4]Kristu Jayanti College, Department of Computer Science (PG), 560077, Karnataka - India
Email: karthik.s@kristujayanti.com - **ORCID:** 0000-0001-7552-8334

Article Info:

DOI: 10.22399/ijcesen.395
Received : 23 July 2024
Accepted : 06 August 2024

Keywords

Classification Algorithms
Internet of Things (IoT)
Machine Learning
Malicious Nodes
Network Security

Abstract:

The prediction of malicious nodes in Internet of Things (IoT) networks is crucial for enhancing network security. Malicious nodes can significantly impact network performance across various scenarios. Machine learning (ML) classification algorithms provide binary outcomes ("yes" or "no") to accurately identify these nodes. This study implements various classifier algorithms to address the problem of malicious node classification, using the "SensorNetGuard" dataset. The dataset, comprising 10,000 records with 21 features, was preprocessed and used to train multiple ML models, including Logistic Regression, Decision Tree, Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Performance evaluation of these models followed the ML workflow, utilizing Python libraries such as scikit-learn, Seaborn, Matplotlib, and Pandas. The results indicated that the Naive Bayes classifier outperformed others with an accuracy of 98.1%. This paper demonstrates the effectiveness of ML classifiers in detecting malicious nodes in IoT networks, providing a robust predictive model for real-time application. The "SensorNetGuard" dataset is available on the IEEE data port and Kaggle platform.

1. Introduction

The smart technology adaptation the Internet of Things (IoT) support forwards the advancements of digital world. The basic principle of IoT infrastructure is connected network of real-time equipment that has the capability to communicate with each other connected physical objects and transfer data to users through the Internet service. The great remarkable growth of IoT in last twenty years is due in part to its wide applicability, scalability, utilization and smartness [1]. The core capable of IoT applications accomplish autonomous tasks in an automated method, with bit or no interference with humans or connected objects. Technology 4.0 is a subclass of IoT, where the terminology associate to the IoT in real-time environments, that is used for saving considerable amounts of valuable resources while increasing productivity revolution in all the fields. Day by day, the proliferation of IoT-enabled services brings an enormous increase in Internet-connected devices [2]. This growth necessitates appropriate safeguards, including security, privacy protection, and policies to mitigate vulnerability potentials, aimed at preventing threats in sectors such as smart industry, smart home, healthcare, logistics, transport, and media, which are at the forefront of the IoT evolution [3][4].

According to cybersecurity researchers, the number of attacks against IoT network devices has increased by more than 100% annually over the past five years. Cybercriminals have adeptly shifted their focus to exploiting IoT infrastructure to steal data and inflict more malicious damage on networks. In response, the cybersecurity research community is developing

significant and sophisticated advanced security tools and techniques to protect data on traditional information systems. However, many of these methods cannot be directly integrated into the IoT environment due to its constrained characteristics network [5].

In an IoT environment, the direct identification of malicious nodes is promptly crucial, and the extensive presence of affected nodes can harm network system functionality [6]. If they are not identified and removed, it will affect the entire network performance at various levels. Malicious node detection and identification are the subjects of numerous proposed works under the development model. However, implementing security mechanisms is difficult due to IoT network nodes being placed far away from the control system. The constrained nature of these nodes (power, size, speed, etc.) does not permit the direct implementation of additional features and different levels of a layered approach [7].

Supervised ML classification algorithms have various applications in our daily lives [8]. Classification algorithms are a type of supervised ML method used to predict the correct label metrics of a given input. In classification, the model data is fully trained on a training dataset and then evaluated the test data before being used for decision-making operations on new datasets. For instance, an algorithm can learn to predict whether a given node is malicious or legitimate based on the trained data.

In a binary classification task, the objective is to categorize input data into one of two distinct groups. The training data for these tasks is labeled in a binary format, usually with 1 indicating "malicious" and 0 indicating "legitimate". The malicious node detection process often requires a straightforward decision-making mechanism either yes (malicious) or no (legitimate). Binary classification algorithms are particularly suitable for this task, as they are designed to provide one of two possible outputs based on the trained dataset.

Effective decision-making problems often involve two mutually exclusive classes, known as binary classification problems [9]. To perform the classification operation, features from the training dataset are used. The classifier evaluates the patterns within the data, with each model having its own set of attributes tailored to the specific task.

1.1 Motivation and Contribution

The motivation behind this performance evaluation fact backtrack stems from the fact that in IoT networks, nodes are randomly located. These randomly located nodes face various issues such as packet loss, retransmission, energy, node lifetime and security risks. In IoT, data is collected live or periodically from various real-time environments such as smart appliances, smart agriculture, and smart industry, among others. Any node can join the existing network without human intervention. Consequently, unauthorized or malicious nodes can easily infiltrate the network and engage in unwanted activities, leading to single points of failure that affect the entire network performance [10].

However, centralized security monitoring and control systems are highly expensive. In many cases, malicious nodes collect false data, deliver data to the wrong destination, and implementing direct security methods may require significant data computation. Furthermore, existing research and proposed methodology has mostly focused on static and smart automated predictive analysis based on bug identification and Intrusion Detection (ID), as well as code issues, with high attention to Artificial Intelligence (AI)-based classification algorithms. However, in the modern digital world, there is a demand to design AI-based smart models to detect malicious nodes in IoT networks using real-time datasets.

1.2 Novelty

In recent digital era, there has been a notable adaptation growth in the use of ML classification algorithms in across various predication domains, such as the weather report, spam mail, loan eligibility, investment, fraud detection, face recognition and so on. Moreover, ML is a subfield of AI that uses categorical value of the output variable to make decisions based on that datasets learning. It is crucial to evaluate malicious node detection model through learning approaches, and essential portion of dataset training and selecting the right real time dataset.

The goal of this evaluation work is to provide an extensive and comprehensive performance of publicly available datasets that can be used for developing the models of IoT network security solutions. In this proposed evaluation analysis work we utilized the publicly available sensor node datasets with 21 features which includes the different real time node metrics like Packet_Drop_Rate, Packet_Duplication_Rate, Data_Throughput, Battery_Level, Is_Malicious and so on. In addition, a detailed explanation of the dataset metrics and its features(fields/columns) is given in this paper later section. Therefore, there is an uncompromising need for improved secure architecture the dataset first classifies and identifies malicious nodes for the model deployment. Towards of achieving our aim of

this analysis model the ML classification algorithms classify the malicious and legitimate nodes using system dataset. Then, the workflow of classification algorithms training and testing the model, fitting the model and evaluation of model proceeded for the selection of algorithms. So, this classifier model evaluating only to choose the right algorithm for malicious node prediction.

1.3 Contribution

The major contribution of this work is as follows:

- In this work, we propose an AI-based model that classifies malicious and legitimate nodes from datasets. The proposed model is organized into an efficient prediction model and utilizes machine learning (ML) classification algorithms.
- We utilized publicly available sensor datasets representing wireless sensor node characteristics and a sufficient amount of dataset features, including node ID.
- We extracted features from various node sources and proposed with 21 features with high correlations. For model preparation, we propose processing and analyzing chosen dataset using ML model workflow steps. This includes conducting primary and fundamental exploratory data analysis and evaluating the performance of machine learning approaches in binary classification algorithms.
- The proposed system evaluation method is evaluated and analyzed using different performance assessment metrics and measures, such as testing and training accuracy, classification measures, and fitting. To better understand the results, this model is implemented in Python.

This paper aims to enhance the security performance of IoT networks by predicting malicious nodes using machine learning classification algorithms. To provide a comprehensive analysis, the paper is organized as follows: Section 2 presents a detailed literature review, summarizing recent advancements and methodologies in the detection of anomalous network traffic and malicious nodes. Section 3 introduces the system model and problem formulation, outlining the theoretical framework and the specific challenges addressed in this study. Section 4 describes the model selection process and evaluates the results obtained from various machine learning classifiers. Finally, Section 5 concludes the paper by summarizing the evaluation performance findings and suggesting future research directions for improving IoT network security.

2. Literature Review

The importance of the Internet of Things (IoT) has revolutionized various application domains by enabling seamless connectivity and data exchange between devices. However, this extensive connectivity also introduces significant security challenges, particularly the threat of malicious nodes that can compromise the integrity and performance of IoT networks. To address these challenges, researchers have extensively explored the application of machine learning (ML) techniques for anomaly detection and security enhancement in IoT environments.

The effectiveness of the FedTrust approach has been extensively evaluated against existing methods in terms of accuracy, precision, and other key metrics. Simulation results demonstrate that FedTrust achieves superior detection and prediction rates for malicious and compromised nodes, highlighting its potential for enhancing IoT network security [11].

The Internet of Things (IoT) has revolutionized connectivity, enhancing automation, productivity, and real-time data utilization and access. However, IoT systems face significant security threats such as malicious nodes, data integrity issues and denial-of-service attacks. Integrating blockchain technology with IoT has shown promise in validating IoT network data through smart contracts, though these can also be highly vulnerable. To address these challenges, the author [12] have proposed AI-based models that detect malicious users using binary classification and employ blockchain for secure data storage. Additionally, deep learning algorithms classify smart contracts to prevent malicious exploitation. These models offer an end-to-end security pipeline, evaluated through metrics like precision, recall, and ROC curves, demonstrating their effectiveness in enhancing IoT security and reliability.

Computer networks face numerous attacks daily, evolving with new methods that target every open port. Tools like network mapping and vulnerability scanning are commonly used to address these threats. Recently, machine learning (ML) has been employed to enhance Intrusion Detection Systems (IDS) by detecting malicious network traffic. The effectiveness and performance of ML models relies on the high quality of the training dataset. The research work [13] proposes an ML-based detection framework for IDS, utilizing the ISOT-CID dataset, which includes both malicious and normal traffic features. By adding six key features, including a novel 'rambling' feature, the study significantly improves anomaly detection accuracy, with DTREE and Random Forest models showing optimal results.

The author [14] employs the publicly available Canadian Institute for Cybersecurity (CIC) IoT dataset to develop machine learning models for efficient detection of anomalous network traffic. The choice of dataset includes thirty-three types of IoT attacks across 7 categories. After pre-processing and balancing the selected dataset, models such as Adaptive Boosting, Logistic Regression, Random Forest, Perceptron, and Deep Neural Network were trained. Random Forest achieved 99.55% accuracy in both multiclass and binary classification models, with reduced computational response time essential for real-time attack detection. The study utilized the SMOTE algorithm for balanced data distribution and demonstrated that Random Forest outperformed other models in all classification strategies.The authors [15] presents a novel method for identifying malicious nodes in WSNs using correlation theory to prevent fault data injection (FDI) attacks. The approach involves detecting anomalies through time correlation, identifying malicious nodes via spatial correlation, and verifying them using event correlation. Experimental comparisons demonstrate that this method achieves better recall result rate and lower false-positive and false-negative rates than standard fuzzy method reputation and weighted-trust-based models. The DDF-2 algorithm was employed to enhance the prediction model and reducing estimation errors. The AdaBoost algorithm was optimized to account for node-fault transmission, and a correlation coefficient calculation was introduced to improve detection accuracy. An event correlation-based detection method further enhanced performance, exemplified by a fire event scenario. Comparative analyses reveal that traditional fuzzy and weighted-trust algorithms lack stability, and credit-based methods overlook the complexity of indicators. The proposed method excels in recall, FPR, and FNR but shows a decline in recall as the proportion of malicious nodes increases, especially beyond 50%. The authors [16] investigates the performance of various machine learning and deep learning algorithms in attack identification systems, specifically using the WMSN-DS database. A Convolutional Neural Network (CNN) combined with Random Forest classifier is proposed for an effective intrusion detection system (IDS) in Wireless Multimedia Sensor Networks (WMSNs). The study highlights deep learning with a Random Forest classifier to identify and avoid attacks, promoting efficient forwarding in WMSNs. Multiple WMSN attacks, including Wormhole, Black hole, Flooding, and TDMA, were critically evaluated. The Random Forest classifier achieved a precision value of 97.00% across all threats. Key metrics used to measure success included recognition effectiveness, false positive rates, false negative rates, and the F1 score.The review of literature highlights the critical role of machine learning algorithms in enhancing the security of IoT networks and WMSNs. Various studies demonstrate the effectiveness of these techniques in detecting and mitigating a wide range of malicious attacks, such as data integrity breaches, denial-of-service attacks, and malicious node identification. Methods such as Random Forest, Convolutional Neural Networks, and hybrid approaches have shown high accuracy and performance in experimental evaluations.However, despite these advancements, there are notable limitations. Many existing models face challenges with scalability and real-time detection due to the high computational requirements. Additionally, the reliability of these models can be compromised by the proportion of malicious nodes in the network, as seen in studies where detection accuracy declines significantly as the number of malicious nodes increases. There is also a need for more robust techniques that can adapt to dynamic changes in the network environment and handle complex node attack scenarios effectively.

3. System Model and Problem Formulation

In this section, for the malicious nodes prediction model design we considered publicly available SensorNetGuard dataset [17] for the performances evaluation and followed basic principles of ML model design. We applied and compared the performances of several supervised algorithms. In the proposed model, we adopted the "SensorNetGuard" dataset to implement various classifiers through a use case for identifying malicious sensor nodes. This dataset comprises 10,000 sample records with twenty-one features and is specifically designed for the identification of malicious nodes in an IoT-based network. All 21 features in this dataset pertain to real-time sensor node data, indicating whether these nodes are malicious or not across various fields. Our goal is to build an ML model to determine whether any newly joining node in the network is malicious. We are using a classification model since the target variable is binary, i.e., 1 (malicious) or 0 (not malicious).

3.1 Datasets Data Collection and Loading

We build the ML based predictive model for finding the if any new joining node in the network is malicious or not depending on the data set values of other features such as Packet_Drop_Rate, Packet_Duplication_Rate, Energy_Consumption_Rate, CPU_Usage Memory_Usage, Data_Transmission_Frequency and so on. We performed steps as per the ML

workflow, which we have discussed in order to make this malicious node predictive analysis and used the sci-kit sklearn, Seaborn, Matplotlib and Pandas library from online Kaggle online platform.

3.2 Understanding the Data

The understanding of the "SensorNetGuard" dimension the Dataframe shape attribute is used to inferences visualize and other methods. In this dataset, we have 10,000 observations with 21 features, here the target value is Is_Malicious and other columns are independent features that will decide the target variable. Then we access the index attribute of the DataFrame, which returns an *Index* object containing the dataset index labels and presented in the Figure 1.

```
print(df.columns)
df.index

Index(['Node_ID', 'Timestamp', 'IP_Address', 'Packet_Rate', 'Packet_Drop_Rate',
       'Packet_Duplication_Rate', 'Data_Throughput', 'Signal_Strength', 'SNR',
       'Battery_Level', 'Energy_Consumption_Rate', 'Number_of_Neighbors',
       'Route_Request_Frequency', 'Route_Reply_Frequency',
       'Data_Transmission_Frequency', 'Data_Reception_Frequency', 'Error_Rate',
       'CPU_Usage', 'Memory_Usage', 'Bandwidth', 'Is_Malicious'],
      dtype='object')
[5]: RangeIndex(start=0, stop=10000, step=1)
```

Figure 1. *Attributes present in SensorNetGuard dataset.*

The info() method is used to understand the data types of each column, and it returns three types of columns in this selected dataset, as presented in Table 1. The object type contains categorical values such as Timestamp and IP_Address. Int64 indicates integer values like Node_ID, Number_of_Neighbors, and Is_Malicious, while the rest of the columns contain float64 data type values among the 21 columns. The countplot() function is used to visualize the comparison of node malicious or not as in our selected dataset. The following Figure 2 could presents the plot out of a total of 10,000 observation, the percentile of 95% observations are there is normal node and not affected and 5% of nodes are malicious from the selected dataset. The 0 shows that on non-malicious nodes and 1 shows that malicious nodes in the selected dataset. As our selected dataset contains many continuous values, we need to transform some of these into categorical attributes for certain machine learning algorithms. For example,

Table 1. *Info() method returned dataset values*

S. No.	Column Name	Dataset Non-Null Count	Data type
1	Node_ID	10000	Int-64
2	Timestamp	10000	object
3	IP_Address	10000	object
4	Packet_Rate	10000	Float-64
5	Packet_Drop_Rate	10000	Float-64
6	Packet_Duplication_Rate	10000	Float-64
7	Data_Throughput	10000	Float-64
8	Signal_Strength	10000	Float-64
9	SNR	10000	Float-64
10	Battery_Level	10000	Float-64
11	Energy_Consumption_Rate	10000	Float-64
12	Number_of_Neighbors	10000	Int-64
13	Route_Request_Frequency	10000	Float-64
14	Route_Reply_Frequency	10000	Float-64
15	Data_Transmission_Frequency	10000	Float-64
16	Data_Reception_Frequency	10000	Float-64
17	Error_Rate	10000	Float-64
18	CPU_Usage	10000	Float-64
19	Memory_Usage	10000	Float-64
20	Bandwidth	10000	Float-64
21	Is_Malicious	10000	Int-64

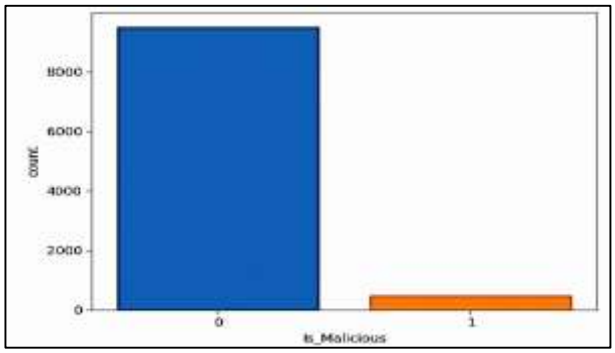

Figure 2. *Dataset distribution of observations based on Is_Malicious.*

we can categorize values like Packet_Drop_Rate, Energy_Consumption_Rate, Packet_Duplication_Rate, CPU_Usage, and Memory_Usage into binary form, such as high (1) or low (0) and dropped the few categorical attributes like Node_ID,Timestamp, IP_Address,SNR, Battery_Level, Bandwidth and Is_Malicious.

This process can help improve the accuracy of predictions. For categorical variables, we apply one-hot encoding, which increases the number of columns in the dataset. After these transformations, the dataset is better prepared for feeding into

machine learning algorithms for accurate result prediction.

3.3 Training and Testing the Model

With the dataset preprocessed, the next step involved building and evaluating the machine learning model. We followed the standard machine learning workflow, which includes splitting the dataset, training the model, and testing its performance. We followed the workflow presented in Figure 3 for the performance evaluation.

Figure 3. *ML model workflow.*

Data Splitting

The chosen dataset was divided into training set and testing set in a 70:30 ratio. This allocation means that 70% of the data was utilized for training the model, while the remaining 30% was used to evaluate its performance. This division helps ensure that the model can generalize effectively to new, unseen data.

Model Training

The training process involved serving the training data into the classification model. During this phase, the model learns the patterns and relationships between the features and the target variable (Is_Malicious).

Model Testing

After training, the model was evaluated on the testing set. This step is crucial to ensure the model's accuracy and its ability to predict whether new nodes joining the network are malicious or not.

Tools and Libraries

The implementation of data pre-processing, testing, training dataset, and machine learning algorithms was conducted by using the scikit-learn library on the Kaggle online platform. The obtained results and code snapshot are presented in Figure 4.

Figure 4. *Training and testing tool result.*

By implementing these steps, we can develop a robust machine learning model to predict whether a new node joining the network is malicious or not.

3.4 Fitting the Logistic Regression Model

We selected Logistic Regression as our classification model due to its effectiveness in binary classification problems. Here are the steps we followed to fit the Logistic Regression model: importing the model, creating an instance of the model, and fitting the model, which is presented in Algorithm 1. In this step, X_train represents the training features(fields), and y_train represents the target variable (malicious node) for the training set. The fit method trains the Logistic Regression model on the provided data.

Algorithm 1. *Model Training and Evaluation*

Input: *Feature matrix_X, target vector_y*
Output: *Accuracy scores of different models*
Step 1 : *Split the Dataset:*
//Logistic Regression Model
Step 2: *Initialize and Train Logistic Regression:*
Step 3: *Fit the model to the training data:*
LRmodel.fit(Xtrain, ytrain)
Step 4: *Predict and Evaluate:*
Step 5: *Compute the accuracy score of the LR model:*
y_predict = model.predict(Xtest)
s = LRmodel.score (Xtest,ytest)
//Print the accuracy score.
display(s)
Step 6: *Generate Confusion Matrix and Classification Report:*
display(a)
//Alternative Models
Step 7: *Support Vector Machine (SVM):*
from sklearn.svm import SVC
svc_model = SVC_model()
svc_model.fit(Xtrain,ytrain)
Step 8: *Decision Tree Classifier:*
from sklearn.tree import DecisionTreeClassifier
dtc_model = DecisionTreeClassifier_model()
dtc_model.fit(Xtrain,ytrain)
Step 9: *Naive Bayes Classifier:*
nb_model =GaussianNB_model()

nb_model.fit(Xtrain,ytrain)

Step 10: *K-Nearest Neighbors (KNN):*
from sklearn.neighbors import KNeighbors Classifier
knn_model =KNeighborsClassifier_model()
knn_model.fit(Xtrain,ytrain)
//Model Performance Evaluation

Step 11: *Print Model Accuracies:*
display(LRmodel.score(Xtest,ytest))
display(dtc_model.score(Xtest,ytest))
display(knn_model.score(Xtest,ytest))
display(svc_model.score(Xtest,ytest))

4. Model Selection and Result Evaluation

We began our model evaluation with Logistic Regression due to its effectiveness in binary classification problems. The Logistic Regression model was trained using the LogisticRegression class from the sklearn.linear_model module, with a maximum of 1000 iterations to ensure convergence.

We prepared our dataset by splitting it into training and testing sets using a 70:30 ratio. This was achieved using the train_test_split function from the sklearn.model_selection module, ensuring consistency with a fixed random state for reproducibility. After training and evaluating our Logistic Regression model, we obtained exceptional classification metrics, as shown in Table 2. The precision, recall, and F1-score for both classes (0 and 1) are all perfect at 1.00, indicating flawless performance with no false positives or false negatives. The overall prediction accuracy of the model is 1.00, meaning it correctly predicted all instances in the test dataset. The macro average and weighted average of precision, recall, and F1-score are also 1.00, demonstrating balanced and accurate predictions across both classes. This result indicates that the Logistic Regression model is highly effective in identifying both malicious and non-malicious nodes in the IoT network dataset.The model demonstrates exceptional performance with perfect precision, recall, and F1-scores for both classes, and an overall accuracy of 100%. This quite indicates that the Logistic Regression model is highly effective in identifying both malicious and non-malicious nodes in the IoT network dataset, without any errors.

4.1 Selection of Model

The performance of each selected model was evaluated based on their accuracy scores on the test dataset. The results are as follows:

These results percentage are summarized in Table 3 and visually represented in Figure 5.Naive Bayes Classifier: This model achieved the notable and highest accuracy of 98.1%, indicating its effectiveness in this context. The Naive Bayes classifier is known for its simplicity and efficiency, especially when feature independence assumptions approximately hold.

Table 2. *Classification Metrics*

Metrics	Precision	Recall	F1-score	Support
Class 0	1.00	1.00	1.00	2844
Class 1	1.00	1.00	1.00	156
Accuracy			1.00	3000
Macro avg	1.00	1.00	1.00	3000
Weighted avg	1.00	1.00	1.00	3000

4.2 Result and Discussion

Logistic Regression: With an accuracy of 97.7%, the Logistic Regression model also performed notably well. Logistic Regression is a robust for binary classification model, offering interpretability and ease of implementation.

Decision Tree Classifier: The Decision Tree model achieved an accuracy of 93.2%. While decision tree models are flexible and capable of capturing non-linear relationships, they are prone to overfitting, which might explain the lower accuracy compared to Naive Bayes and Logistic Regression.

Support Vector Machine (SVM): This SVM model reached an accuracy of 91.3%. SVMs are powerful classifiers, particularly for high-dimensional spaces; however, their performance can be perceptive to the choice of feature and hyperparameters.

Naive Bayes Classifier: This model achieved the highest accuracy of 98.1%, indicating its effectiveness in this context. The Naive Bayes classifier is known for its straightforwardness and efficiency, especially when feature independence assumptions approximately hold.

K-Nearest Neighbors (KNN): The KNN classifier had the lowest accuracy at 84.3%. KNN's performance can be adversely affected by irrelevant features and large datasets due to its instance-based learning approach.The results demonstrate that both the Naive Bayes and Logistic Regression classifiers are highly effective for identifying malicious nodes in an IoT network, with Naive Bayes slightly outperforming Logistic Regression. The high and obtained accuracy of these models suggests they are

well-suited for this binary classification task. The detailed metrics and performance comparisons provided in Table 3 and Figure 5 reinforce the robustness of our evaluation process and highlight the strengths and weaknesses of each classifier.

Table 3. *Model Performance Comparison*

S. No.	Model	Accuracy %
1	Logistic Regression	97.7
2	Decision Tree	93.2
3	Naive Bayes	98.1
4	K-Nearest Neighbors (KNN)	84.3
5	Support Vector Machine (SVM)	91.3

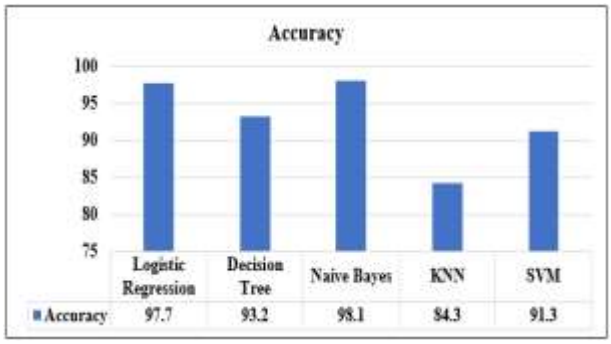

Figure 5. *Accuracy of different models.*

Based on the accuracy scores, the Naive Bayes classifier outperformed the other models with an accuracy of 98.1%, followed closely by the Logistic Regression model with an accuracy of 97.7%. These results indicate that both Naive Bayes and Logistic Regression are effective classifiers for identifying malicious nodes in an IoT network. However, the Naive Bayes model has a slight edge and thus can be considered the best model for this specific dataset and problem.

5. Conclusion

In this article, we applied various machine learning classifiers to identify malicious nodes within an IoT network using a detailed dataset of sensor nodes. Our goal was to assess the performance of classifiers such as Logistic Regression, Decision Tree, Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) in distinguishing between malicious and non-malicious nodes. The Naive Bayes classifier achieved the better highest accuracy of 98.1%, followed closely by Logistic Regression at 97.7%. These results highlight their robustness and suitability for this binary classification task. The third model of Decision Tree classifier model achieved an accuracy of 93.2%, providing valuable interpretability despite slightly lower performance. The SVM model, with 91.3% accuracy, suggests that further hyperparameter tuning could enhance its effectiveness. The KNN classifier had the lowest accuracy at 84.3%, likely due to its sensitivity to irrelevant features and computational complexity.Future work will explore integrating models with IoT platforms for real-time detection and edge computing for on-device anomaly detection. The Naive Bayes and Logistic Regression are effective for identifying malicious IoT nodes, with future enhancements aimed at improving model performance and developing a scalable, real-time intrusion detection system for secure IoT networks.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Toldinas, J., Lozinskis, B., Baranauskas, E., & Dobrovolskis, A. (2019). Mqtt quality of service versus energy consumption. *2019 23rd International Conference Electronics*, 1–4. https://doi.org/10.1109/ELECTRONICS.2019.8765692

[2] Li, B., Ye, R., Gu, G., Liang, R., Liu, W., & Cai, K. (2020). A detection mechanism on malicious nodes in IoT. *Computer Communications*, *151*, 51–59. https://doi.org/10.1016/j.comcom.2019.12.037

[3]Mohamed, K. S. (2019). Iot networking and communication layer. In K. S. Mohamed, *The Era of Internet of Things* (pp. 49–70). Springer International Publishing. https://doi.org/10.1007/978-3-030-18133-8_3

[4] Rane, N., Choudhary, S., & Rane, J. (2023). Artificial Intelligence (Ai) and Internet of Things (Iot) - based sensors for monitoring and controlling in architecture, engineering, and construction: Applications, challenges, and opportunities. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4642197

[5] Ferrag, M. A., Shu, L., Friha, O., & Yang, X. (2022). Cyber security intrusion detection for agriculture 4. 0: Machine learning-based solutions, datasets, and

future directions. *IEEE/CAA Journal of Automatica Sinica*, *9*(3), 407–436. https://doi.org/10.1109/JAS.2021.1004344

[6] Khatun, M. A., Chowdhury, N., & Uddin, M. N. (2019). Malicious nodes detection based on artificial neural network in iot environments. *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 1–6. https://doi.org/10.1109/ICCIT48885.2019.9038563

[7] Prasad, J., Kasiselvanathan, M., Lakshminarayanan, S., Sekar, G., & H, A. (2023). Application of machine learning for malicious node detection in iot networks. *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 1227–1231. https://doi.org/10.1109/IITCEE57236.2023.10091042

[8] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

[9] Smith, M., & Alvarez, F. (2021). A machine learning research template for binary classification problems and shapley values integration. *Software Impacts*, *8*, 100074. https://doi.org/10.1016/j.simpa.2021.100074

[10] Zubair, M., Janicke, H., Mohsin, A., Maglaras, L., & Sarker, I. H. (2024). Automated sensor node malicious activity detection with explainability analysis. *Sensors*, *24*(12), 3712. https://doi.org/10.3390/s24123712

[11] Awan, K. A., Ud Din, I., Zareei, M., Almogren, A., Seo-Kim, B., & Pérez-Díaz, J. A. (2023). Securing iot with deep federated learning: A trust-based malicious node identification approach. *IEEE Access*, *11*,58901–58914. https://doi.org/10.1109/ACCESS.2023.3284677

[12] Shah, H., Shah, D., Jadav, N. K., Gupta, R., Tanwar, S., Alfarraj, O., Tolba, A., Raboaca, M. S., & Marina, V. (2023). Deep learning-based malicious smart contract and intrusion detection system for iot environment. *Mathematics*, *11*(2), 418. https://doi.org/10.3390/math11020418

[13] Alshammari, A., & Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data*, *8*(1), 90. https://doi.org/10.1186/s40537-021-00475-1

[14] Khan, M. M., & Alkhathami, M. (2024). Anomaly detection in IoT-based healthcare: Machine learning for enhanced security. *Scientific Reports*, *14*(1), 5872. https://doi.org/10.1038/s41598-024-56126-x

[15] Lai, Y., Tong, L., Liu, J., Wang, Y., Tang, T., Zhao, Z., & Qin, H. (2022). Identifying malicious nodes in wireless sensor networks based on correlation detection. *Computers & Security*, *113*, 102540. https://doi.org/10.1016/j.cose.2021.102540

[16] John, A. J. S., Roslin, E., & Wilfred, F. (2023). *Deep Learning model-based malicious node detection system in wireless multimedia sensor Network*. https://doi.org/10.21203/rs.3.rs-3066855/v1

[17] Karthick, R., & Arvind, K. S. (2023). *SensorNetGuard: A Dataset for Identifying Malicious Sensor Nodes* [*.csv]. IEEE Dataport. https://doi.org/10.21227/ba0m-cy61