Machine Learning-Driven Anomaly Detection for Enhanced IoT Networks Security

Vijayakumar Ponnusamy[1,*], Siddharth Tiwari[1], Abhinav Sinha[1] and Emilija Kisić[2]

[1]*Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, India*

[2]*Faculty of Information Technology, Belgrade Metropolitan University, Tadeuša Košćuška 63, 11000 Belgrade, Serbia*

Abstract

Although the quick spread of IoT devices has greatly improved operational efficiency and connection across businesses, it has also brought forth serious security flaws. Prior studies on IoT network anomaly detection in smart homes have mostly ignored the possibilities of the Isolation Forest model in favour of methods like Artificial Neural Networks (ANN), Random Forests, and Decision Trees. By developing and assessing the Isolation Forest model to identify anomalies in IoT networks, our work fills this vacuum. The objective of this project is to improve the efficiency and dependability of IoT networks by fortifying them against dynamic cyber threats.

Keywords

Anomaly Detection, IoT network, Cybersecurity

1. Introduction

The Internet of Things (IoT) has rapidly become a cornerstone of modern technology, connecting a vast array of devices across various industries. These devices, ranging from household appliances to industrial sensors, create a network where data flows seamlessly, enabling real-time monitoring, automation, and decision-making. While this interconnection brings great benefits in terms of efficiency, productivity, and user ease, it also exposes IoT devices to major security vulnerabilities. As the number of IoT devices grows, so does the possibility for cyberattacks, data breaches, and operational problems. IoT networks are especially vulnerable to cyberthreats like malware infiltration, unauthorised access, Distributed Denial of Service (DDoS) attacks, and data manipulation. These threats can compromise sensitive data, interrupt services, and even cause physical harm in industrial settings.

A vital security feature, anomaly detection has come to be in order to protect IoT networks from these always changing threats [1]. Anomaly detection techniques, in contrast to conventional methods, which depend on predetermined signatures of known attacks, concentrate on spotting variations from typical network behaviour that may point to unidentified or developing threats [2]. This strategy is especially crucial in dynamic environments such as the Internet of Things (IoT), where communication patterns and devices are ever-changing and static security protocols become less effective. Models for anomaly detection are created to keep an eye on network activity and identify anomalous activities that diverge from standard operating procedures. Examples of these anomalies include sudden requests, anomalous device interactions, or bursts in data transfer.

Artificial Neural Networks (ANN), Decision Trees, Random Forests, and other machine learning methods have been used for anomaly detection in Internet of Things networks [3, 4]. These models have proven to be rather effective at identifying abnormalities, but they frequently demand high processing power and may have trouble processing the huge, high-dimensional datasets that are typical of Internet of Things contexts. Furthermore, a lot of these models are prone to false positives, which are instances

BISEC'2024: 15th International Conference on Business Information Security, November 28-29, 2024, Niš, Serbia

[*] Corresponding author.

vijayakp@srmist.edu.in (V. Ponnusamy); sh5456@srmist.edu.in (S. Tiwari); aa3755@srmist.edu.in (A. Sinha); emilija.kisic@metropolitan.ac.rs (E. Kisić)

000-0002-3929-8495 (V. Ponnusamy); 0000-0003-3059-2353 (E. Kisić)

in which typical behaviours are mistakenly categorised as abnormalities, inefficiently affecting security responses and network monitoring.

The Isolation Tree algorithm, an application of the Isolation Forest model, is a viable substitute for anomaly detection [5, 6]. Recursively partitioning the dataset, this technique isolates data points that seem anomalous by splitting them into smaller segments than normal points. Isolation trees, in contrast to density- or distance-based techniques, concentrate on the ease of isolating a point, which makes them especially useful for identifying outliers in intricate, high-dimensional datasets such as those produced by IoT networks [7, 8]. Because of its scalability and lightweight nature, the Isolation Tree technique is a good fit for real-time anomaly detection in Internet of Things systems. It can handle massive amounts of data with little computing overhead [9].

There are various benefits to utilising the Isolation Tree technique for detecting anomalies in IoT networks. Reducing false positives not only increases detection accuracy but also strengthens the overall security and dependability of IoT networks. This approach aids in proactive threat mitigation by efficiently finding anomalies that might be indicative of cyberthreats or operational issues. The capacity to identify abnormalities in real time is essential for preserving network integrity and guaranteeing the smooth operation of connected devices, given the dynamic nature of cyber threats in IoT environments, where attackers are always coming up with new ways to exploit vulnerabilities. All things considered, the use of the Isolation Tree technique is a positive development for IoT network security. By addressing the shortcomings of conventional anomaly detection techniques, it offers a more effective and efficient way to spot anomalous patterns, strengthening the ability of Internet of Things systems to fend off both known and unknown dangers. As IoT technology develops further, it will be crucial to continuously develop and improve these anomaly detection methods to make sure that the advantages it offers are not outweighed by the rising dangers of cyberattacks and system breakdowns.

2. Ease of Use

The Isolation Tree algorithm offers a high degree of ease of use for anomaly detection in IoT networks, making it an attractive choice for both researchers and industry practitioners. One of its primary advantages is its simplicity in both design and implementation. Unlike more computationally intensive algorithms like Artificial Neural Networks (ANN) or Support Vector Machines (SVM), the Isolation Tree algorithm operates through recursive partitioning, a process that is easy to understand and apply without the need for advanced machine learning expertise. This intuitive process involves isolating anomalous data points based on how easily they can be separated from the rest of the dataset, which aligns well with the nature of IoT data that often includes rare, irregular patterns.

In terms of implementation, the Isolation Tree algorithm does not require extensive tuning of hyperparameters, which can be a significant challenge with more complex models. This reduces the time and effort needed for model configuration, allowing users to quickly deploy the algorithm in real-world IoT environments. Furthermore, the algorithm is compatible with most standard machine learning libraries, making it accessible to those with basic programming knowledge.

Another key aspect of its ease of use is its computational efficiency. Given that IoT networks typically generate massive streams of data, scalability is essential for real-time anomaly detection. The Isolation Tree algorithm is designed to handle large, high-dimensional datasets efficiently, with a low computational overhead. This ensures that the algorithm can process large amounts of IoT data without placing significant strain on system resources, which is especially important in environments with constrained computational power, such as embedded systems or low-power IoT devices.

The interpretability of the Isolation Tree algorithm further enhances its ease of use. The algorithm generates clear, understandable results by isolating anomalous points, allowing users to readily identify and investigate potential threats or system anomalies. Unlike black-box models like deep neural networks, which often provide little insight into how decisions are made, the Isolation Tree algorithm's decision-making process is transparent. This makes it easier for users to validate and trust the results, which is crucial in critical IoT applications such as healthcare, industrial automation, and smart cities, where timely and accurate detection of anomalies can prevent serious disruptions.

Additionally, the algorithm's minimal need for specialized hardware contributes to its broad applicability across various IoT platforms. Many advanced machine learning models require high-performance GPUs or specialized processing units to function effectively, especially when dealing with large datasets. However, the Isolation Tree algorithm can operate on standard computing infrastructure, reducing the cost and complexity of deployment.

Finally, the flexibility of the algorithm allows it to be integrated into various IoT security frameworks with ease. It can be combined with other machine learning techniques or traditional security protocols to provide a layered defense against cyber threats, enhancing its practical value in diverse IoT applications. Its adaptability ensures that it can be applied across different industries, from smart homes and healthcare systems to industrial IoT and connected vehicles, without the need for extensive modification or customization. In summary, the Isolation Tree algorithm's ease of use stems from its simplicity, computational efficiency, scalability, interpretability, and flexibility. These features make it a powerful yet user-friendly tool for improving anomaly detection in IoT networks, enabling enhanced security with minimal technical barriers for adoption.

2.1. Equations

In the context of this study, the Isolation Forest algorithm is applied to detect anomalies within IoT network data, leveraging the fundamental concept that anomalies are more easily isolated than normal points. The following mathematical formulations and principles are used to understand and quantify the behavior of the Isolation Forest model.

1. Path Length Calculation
   For a data point x, the path length $h(x)$ is defined as the number of edges traversed from the root of an isolation tree to the leaf node where x is located. Since anomalies are isolated more quickly than normal points, the path length provides an indication of how anomalous a point is. The average path length for a point x in an isolation tree built from a sample of n points can be approximated by:

   $$h\left(x\right) \approx 2H\left(n-1\right) - \left(\frac{2\left(n-1\right)}{n}\right), \tag{1}$$

   where $H(n)$ is the harmonic number, which approximates the average path length in a completely random tree. It can be computed as $H(n) = ln(n) + y$, where $y \approx 0.577$ is Euler's constant, and n is the number of data points in the tree.
2. Anomaly Score Calculation
   The anomaly score $s(x, n)$ for a point x is computed based on the path length. The shorter the path length, the higher the likelihood that the point is an anomaly. The anomaly score is given by:

   $$s(x, n) = 2^{-c(n)E(h(x))}, \tag{2}$$

   where $E(h(x))$ is the average path length of x across all isolation trees in the forest, and $c(n)$ is the average path length of a point in a binary tree built from n samples, and is given by

   $$c\left(n\right) = 2H\left(n-1\right) - \left(\frac{2\left(n-1\right)}{n}\right). \tag{3}$$

   The anomaly score $s(x, n)$ ranges between 0 and 1, with higher values indicating that the point is more likely to be an anomaly:
   $s(x, n) \to 1$ implies a high likelihood of the point being an anomaly;
   $s(x, n) \approx 0.5$ indicates the point is typical;
   $s(x, n) \to 0$ implies the point is likely normal.

3. Related Works

Recent advancements in anomaly detection for IoT networks have produced a variety of methodologies aimed at enhancing security, operational efficiency, and system reliability. One significant approach involves the use of Sparse Autoencoders (SAEs) for dimensionality reduction, followed by the application of Convolutional Neural Networks (CNNs) for effective anomaly detection. This SAE-CNN framework has shown promising capabilities in identifying unusual patterns within network traffic. However, the inherent complexity of the model raises challenges for real-time detection, particularly in resource-constrained IoT environments where computational power and memory are limited. Additionally, the validation of this model solely on a single dataset, such as the Bot-IoT dataset, raises concerns regarding its generalizability to the diverse and dynamic nature of real-world IoT network traffic. This limitation highlights the need for further validation across multiple datasets to ensure robustness [10].

Another noteworthy study conducted a thorough analysis of a different IoT dataset using several machine learning classifiers, including Random Forest, Decision Tree, AdaBoost, and Artificial Neural Networks (ANN) [11, 12]. The evaluation of these models was grounded in metrics such as weighted precision, recall, and F1 scores, providing a nuanced understanding of their performance. Despite this comprehensive evaluation, the models displayed limited interpretability when deployed in varied environments. This lack of transparency could hinder practical implementations, especially in critical applications where understanding model decisions is vital. Moreover, the study identified scalability issues when incorporating additional datasets, which can complicate the integration of new data sources. There is also a pressing need for unsupervised techniques that could bolster model generalization for new and unlabeled data, thereby improving the adaptability of these models in real-world settings [13].

In addition to these methodologies, a systematic review of studies focusing on anomaly detection in industrial machinery utilizing IoT devices offered valuable insights into various aspects, including the types of machinery employed, the sensors used for data collection, and the preprocessing methods applied [14]. This review synthesized findings from numerous studies, highlighting the importance of understanding the interplay between different factors affecting anomaly detection performance. However, the narrow focus on recent literature limited the exploration of foundational research, potentially overlooking valuable insights that could inform current practices. The variability in sensors and machinery across studies resulted in inconsistent preprocessing techniques, complicating comparative analyses. Furthermore, the diversity of machine learning algorithms applied in different studies presents challenges for drawing clear conclusions and establishing best practices in anomaly detection.

Moreover, several studies have underscored the necessity for interdisciplinary approaches, integrating domain knowledge from fields such as cybersecurity, data science, and engineering to improve anomaly detection systems. This integration could facilitate the development of more sophisticated models capable of adapting to evolving threats in IoT environments. The overall quality of the studies reviewed varies significantly, which raises questions about the reliability and applicability of the findings in practical scenarios. The need for standardized evaluation frameworks and benchmarks in the field of IoT anomaly detection is evident, as it would enhance the comparability of research outcomes and facilitate the development of more effective detection systems [15].

In conclusion, while the body of research surrounding anomaly detection in IoT networks is expanding rapidly, challenges remain in terms of model complexity, interpretability, scalability, and the need for a deeper understanding of the diverse operational environments in which these models will be deployed. Addressing these challenges will be crucial for advancing the effectiveness of anomaly detection systems and ensuring their reliability in real-world applications.

4. Methodology

This study employs the Isolation Forest model to detect anomalies in IoT networks, leveraging the BOT-IoT dataset obtained from Kaggle. The methodology comprises several crucial steps: dataset acquisition, data preprocessing, feature engineering, model training, and evaluation.

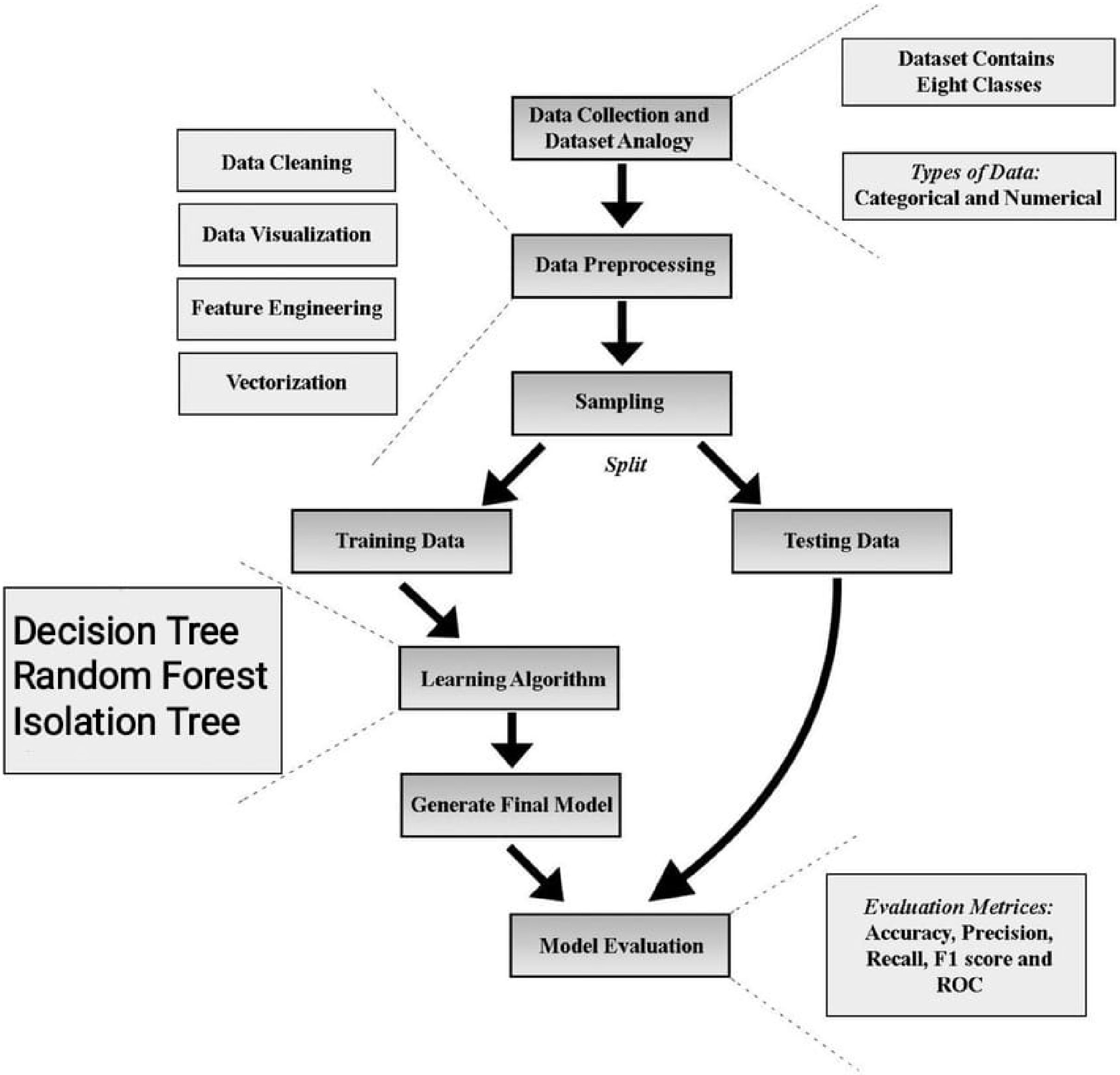

Figure 1: Proposed system.

4.1. Dataset Acquisition

The BOT-IoT dataset was sourced from Kaggle, comprising a comprehensive collection of IoT network traffic data. This dataset contains various records of network interactions, including normal behavior and several types of attacks. It serves as an ideal benchmark for evaluating anomaly detection algorithms, specifically in IoT environments

4.2. Data Inspection

Upon loading the dataset, we conducted an initial exploratory data analysis (EDA) to understand its structure, dimensions, and characteristics. Key attributes such as destination port (dport) and bytes transferred were examined to identify unique values and potential data anomalies. This step also involved assessing the distribution of various features to recognize patterns that might influence the model's performance.

4.3. Data Cleaning

Data cleaning was an essential phase to ensure the quality and reliability of the dataset. We undertook the following actions:

- Data Type Conversion: Relevant columns, particularly ’dport’ and ’bytes,’ were converted to numeric formats. This conversion is crucial since machine learning algorithms require numerical inputs for processing.
- Handling Missing Values: After conversion, we checked for NaN (Not a Number) entries within critical columns. Rows containing NaN values were dropped to avoid complications during model training. Alternatively, other strategies, such as filling NaN values with a default constant, were considered but ultimately not implemented in this instance.
- Removing Unnecessary Columns: Columns that provided little value to the model, such as MAC addresses and unnecessary identifiers, were removed to streamline the dataset and focus on the most relevant features for anomaly detection.

4.4. Categorical Encoding

As the dataset included categorical features, it was necessary to convert these variables into a numerical format to facilitate the learning process. Label encoding was applied to categorical attributes such as ’flgs’ (flags), ’proto’ (protocol), ’saddr’ (source address), ’daddr’ (destination address), ’state’, ’category’, and ’subcategory’. Each category was transformed into a unique integer, allowing the Isolation Forest algorithm to process these features effectively.

4.5. Feature Definition

The model’s performance heavily depends on the features selected for training. We defined our feature set by excluding the target columns, particularly ’attack’ and ’category’, which are used as labels for classification. The final feature set comprised several numerical and categorical attributes that provide insights into network behavior.

4.6. Feature Engineering

To enhance the model’s predictive power, we created a new feature termed "bytes per packet." This feature was calculated by dividing the total bytes transferred by the number of packets for each record. The introduction of this feature is vital, as it offers a more nuanced perspective on network activity, aiding the model in distinguishing between normal and anomalous traffic patterns.

4.7. Data Normalization

Given the diverse ranges of the features, normalization was performed to ensure that each feature contributed equally to the model’s learning process. We applied standardization, which rescales the features to have a mean of zero and a standard deviation of one. This step is particularly important for algorithms like Isolation Forest, where the distance metric plays a crucial role in anomaly detection.

4.8. Train-Test Split

To evaluate the model’s performance accurately, the dataset was split into training and testing sets using an 80-20 ratio. The training set comprised 80% of the data, which the model would learn from, while the remaining 20% served as the test set for validation. This separation is critical to prevent overfitting and ensure that the model generalizes well to unseen data.

4.9. Model Initialization and Training

The Isolation Forest model was initialized with specific hyperparameters, including the number of estimators (`n_estimators`), the maximum samples to be drawn (`max_samples`), and the contamination rate (`contamination`) to define the expected proportion of anomalies in the data. After initialization, the model was trained using the training dataset. The training process involves constructing an ensemble of isolation trees, where each tree partitions the feature space recursively until it isolates observations, thereby identifying anomalies based on their average path lengths in the trees.

4.10. Prediction and Evaluation

Once trained, the model was used to make predictions on the test dataset. The output of the model includes a binary classification: -1 for anomalies and 1 for normal observations. To facilitate interpretation, the predictions were converted to a binary format, where 1 indicates an anomalous instance and 0 denotes a normal instance.

The model's performance was evaluated using various metrics, including precision, recall, F1-score, and confusion matrix analysis. These metrics provide insights into the model's accuracy in identifying anomalies, balancing false positives and false negatives to assess overall effectiveness.

5. Results

The performance of the Isolation Forest model in detecting anomalies in IoT networks was systematically evaluated using several metrics, including accuracy, precision, recall, F1 score, and a detailed analysis of the confusion matrix. Each of these metrics provides insights into the model's ability to accurately classify network traffic and its effectiveness in identifying potential threats.

1. Accuracy
   The Isolation Forest model achieved an overall accuracy of 82.5%. This indicates that the model correctly identified 82.5% of instances across the dataset, signifying a strong capacity to differentiate between normal and anomalous traffic. High accuracy is particularly important in IoT environments where network reliability and security are paramount. It reflects the model's ability to learn from the underlying patterns in the data, suggesting that it can effectively generalize its understanding to unseen instances.
2. Precision
   Precision, calculated at 97.27%, illustrates the model's proficiency in identifying true anomalies among the instances it predicted as anomalous. A high precision value indicates that when the model flags an instance as anomalous, it is very likely to be correct, thus minimizing the impact of false positives. In practical applications, this is crucial, as false alarms can lead to unnecessary alerts, wasted resources, and potential disruption of normal operations. This high precision suggests that the model is well-suited for environments where the cost of false alarms is significant, such as critical infrastructure and industrial settings.
3. Recall
   The recall metric, standing at 87.4%, indicates the proportion of actual anomalies correctly identified by the model out of all true anomalies present in the dataset. While an 87.4% recall is commendable, it also highlights the model's inability to detect some anomalies (false negatives), which could represent missed security threats or system failures. This aspect emphasizes the necessity for continuous improvement in recall, as high recall is vital for a robust security framework, ensuring that most potential threats are identified and addressed promptly.
4. F1 Score
   The F1 score of 94.4% serves as a balanced measure that combines both precision and recall. This score is particularly relevant in the context of imbalanced datasets, common in anomaly detection tasks where normal instances typically outnumber anomalous ones. A high F1 score signifies that the model not only has strong precision but also demonstrates considerable recall, indicating

effective identification of both normal and anomalous traffic. This balance is essential for the model's reliability in operational environments, where both types of misclassifications (false positives and false negatives) can lead to significant consequences.

5. Confusion Matrix The confusion matrix provides a granular view of the model's classification performance. Analyzing the confusion matrix reveals important insights:
   - True Negatives (TN): The model accurately identified 855 instances as normal, reflecting its ability to recognize benign traffic patterns.
   - False Positives (FP): There were 225 instances incorrectly classified as anomalies, indicating the potential for unnecessary alerts that could lead to operational inefficiencies. This suggests that further tuning of the model is needed to refine its sensitivity to normal traffic.
   - False Negatives (FN): A substantial 153,397 normal instances were misclassified as anomalies, highlighting a significant area for improvement. This emphasizes the need for further investigation into the features contributing to this misclassification and potential enhancements in feature selection or model architecture.
   - True Positives (TP): The model successfully detected 8,005 actual anomalies, confirming its utility in identifying genuine threats. This high number of detected anomalies is promising, as it indicates the model's capacity to safeguard against various attack vectors.

6. Discussions

The results obtained from the evaluation of the Isolation Forest model reveal significant insights into its effectiveness and limitations for anomaly detection in IoT networks. The high accuracy, precision, recall, and F1 score illustrate the model's capability to discern between normal and anomalous traffic. However, a closer examination of the confusion matrix and the underlying factors contributing to these results provides a nuanced understanding of the model's performance.

1. Implications of High Precision and F1 Score
   The achieved precision of 97.27% indicates that the model has a strong ability to identify true anomalies without overwhelming the system with false alarms. In practical terms, this is particularly valuable in operational settings where the cost of false positives can lead to unnecessary downtime, resource allocation, and potential disruptions. The high F1 score of 94.4% further emphasizes that the model maintains a balanced performance across both precision and recall. This balance is critical in real-time applications, as it demonstrates that the model can effectively handle the imbalanced nature of the dataset, where anomalous instances are often significantly fewer than normal ones.
2. Addressing the False Negatives
   Despite the strengths of the model, the presence of 153,397 false negatives is a concerning aspect of the results. Each false negative represents a missed opportunity to detect an actual anomaly, which could result in undetected attacks or system failures. In IoT environments, where devices may be interconnected and autonomous, the implications of such oversights can be dire. This highlights the need for strategies aimed at improving recall. Potential approaches include:
   - Feature Selection and Engineering: Investigating additional features that capture more nuanced aspects of network behavior can enhance the model's sensitivity. For instance, analyzing temporal patterns or incorporating environmental context could lead to better detection of anomalies.
   - Model Ensemble Techniques: Exploring ensemble methods that combine multiple algorithms might improve overall detection capabilities. For instance, integrating the Isolation Forest with supervised learning methods could help leverage labeled data for more accurate predictions.
   - Threshold Adjustment: Modifying the threshold for classifying an instance as an anomaly can also be a straightforward yet effective approach. By adjusting the decision boundary,

the model may be able to capture more anomalies while maintaining an acceptable level of false positives.

3. Generalizability and Robustness
   The evaluation metrics reflect the model's ability to generalize its learning to the unseen test dataset. However, the training was conducted solely on the BOT-IoT dataset, which raises questions about the model's robustness in diverse operational environments. Real-world IoT networks often exhibit a variety of traffic patterns influenced by different devices, applications, and network configurations. Therefore, training the model on a more diverse set of datasets that incorporate varying types of attacks, protocols, and normal behaviors is essential. Such an approach could enhance the model's adaptability to various IoT scenarios, increasing its effectiveness in live environments.
4. Real-World Applications and Considerations
   The promising results of the Isolation Forest model suggest its practical applicability in securing IoT networks. Organizations aiming to implement anomaly detection systems can consider this model as a foundational component of their cybersecurity strategy. However, deploying such models in real-world settings requires a comprehensive understanding of the specific IoT environment. Key considerations include:
   - Resource Constraints: IoT devices often operate with limited computational power and energy resources. The model's complexity must be balanced against the available resources to ensure real-time detection without overwhelming the devices.
   - Adaptability to Evolving Threats: As cyber threats continue to evolve, the model should be periodically retrained and validated with new data to maintain its effectiveness. Continuous monitoring and updating the detection model will be essential to address new attack vectors and tactics.
   - Integration with Existing Security Frameworks: The anomaly detection model should be integrated into a broader security framework that includes other defensive measures, such as firewalls and intrusion prevention systems. This layered security approach can enhance overall network resilience.
5. Future Directions for Research The results of this study provide a solid foundation for future research in the domain of anomaly detection for IoT networks. Areas for further investigation include:
   - Investigating Hybrid Models: Exploring the combination of supervised and unsupervised learning techniques could lead to improved anomaly detection capabilities, allowing for a more comprehensive understanding of normal and anomalous behavior.
   - Deep Learning Approaches: While the Isolation Forest model demonstrates significant promise, investigating deep learning techniques such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs) for anomaly detection could provide additional insights and improvements.
   - Real-Time Data Processing: Developing frameworks for real-time data processing and anomaly detection will be crucial for the deployment of such models in dynamic IoT environments. This will involve optimizing the model's efficiency to ensure timely responses to detected anomalies.

7. Conclusions

In summary, while the Isolation Forest model shows considerable promise in detecting anomalies within IoT network traffic, the challenges presented by false negatives necessitate ongoing refinement and enhancement. The insights gained from this study not only highlight the model's effectiveness but also underscore the need for continued research and development to bolster its performance and adaptability in real-world scenarios. The commitment to refining these models is essential to ensure the security and reliability of IoT networks, particularly as they continue to expand and evolve in complexity.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] N. Sarwar, I. S. Bajwa, M. Z. Hussain, M. Ibrahim, K. Saleem, Iot network anomaly detection in smart homes using machine learning, IEEE Access (2023).

[2] C. Koetsier, J. Fiosina, J. N. Gremmel, J. P. Müller, D. M. Woisetschläger, M. Sester, Detection of anomalous vehicle trajectories using federated learning, ISPRS Open Journal of Photogrammetry and Remote Sensing 4 (2022) 100013.

[3] S. F. Chevtchenko, E. D. S. Rocha, M. C. M. Dos Santos, R. L. Mota, D. M. Vieira, E. C. De Andrade, D. R. B. De Araújo, Anomaly detection in industrial machinery using iot devices and machine learning: A systematic mapping, IEEE Access 11 (2023) 128288–128305.

[4] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE signal processing magazine 37 (2020) 50–60.

[5] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[6] T. Wang, B. Zhao, L. Fang, Flforest: Byzantine-robust federated learning through isolated forest, in: 2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2023, pp. 296–303.

[7] P. J. Rousseeuw, M. Hubert, Robust statistics for outlier detection, Wiley interdisciplinary reviews: Data mining and knowledge discovery 1 (2011) 73–79.

[8] A. Smiti, A critical overview of outlier detection methods, Computer Science Review 38 (2020) 100306.

[9] A. A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for iot time-series data: A survey, IEEE Internet of Things Journal 7 (2019) 6481–6494.

[10] S. Haider, A. Abbas, A. K. Zaidi, A multi-technique approach for user identification through keystroke dynamics, in: Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0, volume 2, IEEE, 2000, pp. 1336–1341.

[11] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407 (2019).

[12] S. J. Rigatti, Random forest, Journal of Insurance Medicine 47 (2017) 31–39.

[13] J. Zhang, M. Zulkernine, A. Haque, Random-forests-based network intrusion detection systems, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38 (2008) 649–659.

[14] G. Pang, C. Shen, L. Cao, A. V. D. Hengel, Deep learning for anomaly detection: A review, ACM computing surveys (CSUR) 54 (2021) 1–38.

[15] N. K. Sahu, I. Mukherjee, Machine learning based anomaly detection for iot network:(anomaly detection in iot network), in: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), IEEE, 2020, pp. 787–794.