Exploring the effectiveness of Transformer models in detecting IOT Security issues from developer discussions

Nithin. G [1], Vinay. K [2], Thiruumalesh. N [3], Mr. Nagavenkateshwara Rao. K [4]
[1,2,3] UG Scholar, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana,India-500100
[4]Assistant Professor, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India-500100
ganganithin487@gmail.com

Abstract:

In recent years, the Internet of Things (IOT) has grown exponentially, connecting devices to exchange data and automate processes across various industries. However, this rapid growth has led to significant security challenges, particularly vulnerabilities discussed in developer forums. Historically, security measures in IoT focused on static analysis and rule-based systems, which often failed to adapt to evolving security threats. With the rise of Artificial Intelligence (AI), Machine Learning have been introduced to analyse developer discussions and detect security issues more effectively. The objective of this study is to develop a machine learning-based approach, to automatically detect and classify IOT security issues based on developer discussions. This will help in improving the efficiency and accuracy of identifying potential vulnerabilities and security threats in IOT systems.

Keywords: Artificial Intelligence, Machine Learning, Internet Of Things , Security issues Detection.

1. INTRODUCTION

The increasing adoption of the Internet of Things (IoT) has revolutionized industries by enabling smart connectivity, automation, and data-driven decision-making. However, as the number of IoT devices continues to grow, so do the security threats associate with them. These devices often have limited security mechanisms, making them vulnerable to cyberattacks, unauthorized access, and data breaches. As a result, ensuring the security of IoT systems has become a critical challenge for researchers, developers, and security analysts. One of the key sources of information regarding IoT security issues is developer discussions on platforms like Stack Overflow, GitHub, and security forums. Developers frequently discuss vulnerabilities, misconfigurations, and attack mitigation strategies in these forums. However, manually analysing large volumes of developer discussions to identify security-related issues is a time-consuming and inefficient process. Traditional keyword-based searches and rule-based filtering methods often fail to capture the context and evolving nature of security threats, leading to inaccurate or incomplete security issue detection. To address the challenges, recent advancements in Natural Language Processing (NLP), particularly Transformer models, have shown significant potential in automating security issue detection from unstructured text data.
Transformer-based architectures, such as BERT

(Bidirectional Encoder Representations from Transformers), RoBERTa, and GPT, have demonstrated superior performance in understanding complex language structures and extracting contextual information. By leveraging these models, it is possible to accurately identify security vulnerabilities, attack patterns, and risk factors from developer discussions. The research aims to explore the effectiveness of Transformer models in detecting IoT security issues from developer discussions. By analysing textual data using advanced NLP techniques, the study seeks to improve the automation and accuracy of security issue identification. The findings of this study will contribute to the development of AI-driven security monitoring tools, enhancing the proactive detection and mitigation of IoT security threats.

The Internet of Things (IoT) is rapidly expanding across industries, enabling smart devices and interconnected systems to enhance automation and efficiency. However, this widespread adoption has also led to an increase in security vulnerabilities, making IoT devices attractive targets for cyberattacks. Many of these security risks stem from misconfigurations, weak authentication, unpatched software, and insecure communication protocols. Detecting and mitigating such vulnerabilities is crucial for ensuring the reliability, confidentiality, and integrity of IoT ecosystems. One of the primary sources of security insights is developer discussions on platforms such as Stack Overflow, GitHub, and security forums. Developers and security experts frequently report, discuss, and attempt to resolve various IoT security concerns. However, these discussions generate massive amounts of unstructured textual data, making it challenging for manual methods to effectively extract and classify security-related issues. Traditional keyword-based searches and rule-based filtering techniques are inadequate as they often fail to capture context, intent, and evolving security threats.

2. LITERATURE SURVEY

Bidirectional Encoder Representations from Transformers (BERT), first introduced by Jacob Devlin et al., is a transformer model designed to represent language. It was created in such a way that it can be fine-tuned with an additional layer to develop new models that address a wide array of tasks. Yinhan Liu et al. introduced RoBERTa as a replication study of BERT. The authors demonstrated that their model was better trained than BERT, overcoming its limitations and showing superior performance compared to the basic BERT model. Zhilin Yang et al. introduced XLNet, which addresses BERT's constraints by using a universal autoregressive pre-training technique. The maximum expected likelihood was considered over all factorization order arrangements. BERT Overflow, created by Jeniya Tabassum et al., is a type of transformer model trained to identify code tokens or software-oriented entities within natural language sentences. Uddin et al. aimed to learn about the practical difficulties developers face when building real IoT systems. Recent studies explored the connection between API usage and Stack Overflow discussions, revealing a relationship between API class use and the number of Stack Overflow questions answered. Ahmed et al. compared their proposed Transformer architecture with RNN and LSTM for binary classification using the ToN_IoT dataset released in 2020. The results indicated that the proposed Transformer model performs excellently in terms of accuracy and precision, achieving an accuracy rate of 87.79%. He et al. proposed a transferable and adaptive network intrusion detection system (NIDS) based on deep reinforcement learning, achieving 99.60% and 95.60% accuracy in binary and multi-class classification of the CIC-IoT2023 dataset, respectively. Jony et al. used LSTM to conduct an experimental evaluation of multi-class classification in CIC-IoT-2023, achieving an accuracy rate of 98.75%. Jaradat et al. used four different machine learning methods to classify network attacks in CIC-IoT-2023, but did not specify the classification tasks used. Abbas et al. addressed the problem of data imbalance in the dataset and summarized the key points of the aforementioned papers. The effectiveness of machine learning-based intrusion detection systems (ML-IDSs) largely depends on the quality of the dataset.

The provided image presents a literature survey focused on the application of machine learning and deep learning techniques in network intrusion detection systems (NIDS), particularly within the context of the Internet of Things (IoT). The survey highlights various studies that leverage transformer models, recurrent neural networks (RNNs), and other machine learning algorithms to address the challenges of detecting and classifying network attacks.

Bidirectional Encoder Representations from Transformers (BERT) and its Variants

Introduced by Jacob Devlin et al., BERT is a transformer-based model designed for natural language processing tasks. Its ability to be fine-tuned for various applications has led to its adoption in network security. Yinhan Liu et al. presented RoBERTa as an enhanced version of BERT, demonstrating superior performance. Additionally, Zhilin Yang et al. introduced XLNet, which addresses BERT's limitations through an autoregressive pre-training technique.

Transformer Models for Network Intrusion Detection

Jeniya Tabassum et al. developed BERT Overflow, a transformer model trained to identify code tokens and software-oriented entities within natural language sentences. This approach is relevant to detecting software-based attacks. Ahmed et al. compared a custom transformer architecture with RNNs and LSTMs for binary classification using the ToN_IoT dataset, achieving an accuracy rate of 87.79%. This study underscores the potential of transformer models in network security.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks

RNNs and LSTMs are also explored for NIDS. He et al. proposed a transferable and adaptive NIDS based on deep reinforcement learning, achieving high accuracy rates (99.60% and 95.60%) in binary and multi-class classification using the CIC-IoT2023 dataset. Jony et al. utilized LSTM for multi-class classification in the same dataset, achieving an accuracy rate of 98.75%. These studies highlight the effectiveness of RNNs and LSTMs in detecting network intrusions.

Machine Learning Algorithms for Network Attack Classification

Jaradat et al. employed four different machine learning methods to classify network attacks in the CIC-IoT2023 dataset. While the specific classification tasks were not detailed, this study indicates the applicability of various machine learning algorithms in this domain. Abbas et al. addressed the issue of data imbalance in the dataset and summarized key findings from the aforementioned papers. . Their work emphasizes the importance of data quality in machine learning-based intrusion detection systems.

Challenges and Future Directions

The literature survey reveals that the effectiveness of machine learning-based intrusion detection systems largely depends on the quality of the dataset. Data imbalance and the need for robust, adaptable models remain significant challenges. Future research could focus on developing more sophisticated transformer architectures, improving the generalization capabilities of deep learning models, and addressing the limitations of current datasets. Additionally, exploring novel techniques for feature engineering and data augmentation could enhance the performance of NIDS.

In conclusion, the literature survey provides a comprehensive overview of the current state of research in applying machine learning and deep learning to network intrusion detection, particularly in the context of IoT. It highlights the strengths and weaknesses of various approaches and suggests directions for future research.

3. PROPOSED METHODOLOGY

The project explores the effectiveness of Transformer models in detecting IoT security issues by analysing developer discussions. Traditional methods rely on rule-based filtering, manual audits, and signature-based approaches, which often fail to identify new and evolving threats. With the rise of large-scale IoT deployments, security vulnerabilities have become a critical concern, requiring advanced AI-driven solutions. By leveraging Natural Language Processing (NLP) and deep learning, particularly Transformer-based models (e.g., BERT, GPT, or RoBERTa), this project aims to automate and enhance the detection of security-related issues in developer discussions, forums, and bug reports. These discussions often contain valuable insights into security flaws, potential vulnerabilities, and mitigation strategies.

OBJECTIVES

1. **Develop an AI-driven model** that utilizes Transformer architectures for **context-aware detection** of IoT security issues.
2. **Analyse developer discussions** from various sources (e.g., Stack Overflow, GitHub, forums) to **identify security vulnerabilities**.
3. **Improve detection accuracy** compared to traditional rule-based or keyword-matching methods.
4. **Automate security issue classification** into relevant categories for better security monitoring.
5. **Evaluate model performance** against existing security detection techniques using precision, recall, and F1-score metrics.
6. **Enhance IoT security monitoring** by integrating Transformer-based insights into security frameworks.

PROPOSED WORKFLOW

The proposed workflow for detecting IoT security issues using Transformer models follows a structured pipeline that includes data collection, preprocessing, model training, evaluation, and insights generation. The process starts with data collection, where developer discussions from sources like Stack Overflow, GitHub issues, and security forums are gathered. These discussions offer valuable insights into IoT security vulnerabilities, reported issues, and potential mitigations. After collecting the data, text preprocessing is performed, which includes tokenization, stop word removal, lemmatization, and feature extraction using techniques like TF-IDF and word embeddings. This step ensures that the textual data is cleaned and structured for analysis. Next, a Transformer-based model (e.g., BERT, RoBERTa, or GPT) is trained on the processed text data. The model learns contextual relationships and patterns related to IoT security vulnerabilities. A labelled dataset is used for supervised learning, enabling the model to classify and detect potential security issues with high accuracy. Advanced fine-tuning techniques are applied to enhance performance, and the model undergoes multiple iterations to optimize hyperparameters. Following training, the model is evaluated using performance metrics such as precision, recall, F1-score, and accuracy. These metrics help assess how effectively the model detects IoT security issues compared to traditional machine learning models (e.g., SVM, Decision Trees, or Random Forest).

FLOW CHART OF CODE

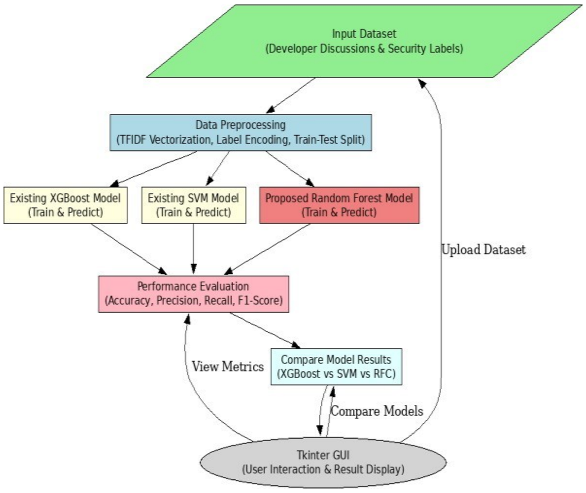


Fig. 1 Flow chart

MODEL BUILDING & TRAINING

The model-building and training process for detecting IoT security issues from developer discussions involves multiple steps, including data preprocessing, feature extraction, model selection, training, evaluation, and optimization. In this project, we employ three key machine learning models: XGBoost (XGB), Support Vector Machine (SVM), and Random Forest Classifier (RFC).

1. Data Preprocessing and Feature Extraction

Before training the models, the dataset undergoes preprocessing. The raw textual data from developer discussions is cleaned using techniques such as tokenization, stopword removal, lemmatization, and normalization. Since machine learning models require numerical input, the text data is transformed into vectorized representations using TF-IDF (Term Frequency-Inverse Document Frequency).

2. Splitting the Dataset

The dataset is split into training and testing sets using an 80-20 ratio. This ensures that the models are trained on a majority of the data while retaining a portion for evaluation.

3. Training Different Machine Learning Models

1. **XGBoost (Extreme Gradient Boosting - XGB)**
   - ○ XG Boost is an ensemble learning method that builds multiple decision trees to enhance classification accuracy.
   - ○ The model is trained with 100 estimators, a learning rate of 0.1, and a maximum depth of 5 to ensure a balance between complexity and performance.
   - ○ After training, it predicts security issue labels based on developer discussions.
2. **Support Vector Machine (SVM)**
   - ○ SVM is a supervised learning algorithm that classifies data points by finding the optimal decision boundary (hyperplane).
   - ○ We use the poly kernel with high regularization (C=1e10) and low gamma (1e-10) for effective text classification.
   - ○ The trained SVM model classifies whether a discussion contains an IoT security issue or not.
3. **Random Forest Classifier (RFC)**
   - ○ RFC is an ensemble learning model based on multiple decision trees, reducing overfitting and improving generalization.
   - ○ The model is trained on the extracted features and optimized using grid search to find the best hyperparameters.
   - ○ Predictions are made by aggregating results from multiple decision trees.

4. Model Evaluation

After training, all models are evaluated using key performance metrics, including:

- **Accuracy** – Measures overall correctness.
- **Precision** – Measures how many detected security issues are actual issues.
- **Recall** – Measures the model’s ability to detect all security issues.
- **F1-Score** – Balances precision and recall.
- **Confusion Matrix** – Visualizes the model’s performance in classifying "Yes" and "No" security issues.

4. EXPERIMENTAL ANALYSIS

IMPLEMENTATION DESCRIPTION

The Python program uses Tkinter to create a graphical user interface (GUI) for detecting IoT security issues in developer discussions using machine learning models. Here's an overview of how it works:

1. Dataset Upload & Preprocessing

- The user uploads a dataset containing developer discussions labeled with security issues.
- The dataset is preprocessed by extracting relevant columns, encoding security issue labels, and converting text data into numerical features using **TF-IDF (Term Frequency-Inverse Document Frequency)**.
- The dataset is split into training and testing sets.

2. Model Training with XGBoost, SVM, and RFC

XGBoost (XGB) Model

- XGBoost is an **ensemble learning method** that uses **gradient boosting** to improve model performance.
- It builds multiple weak decision trees sequentially, optimizing each iteration to correct previous errors.
- Key hyperparameters such as **learning rate, number of estimators, and tree depth** are tuned to enhance accuracy.

Support Vector Machine (SVM) Model

- SVM is a **supervised learning algorithm** that finds the best hyperplane to separate data into different categories.
- The model is trained using different **kernel functions** (linear, polynomial, or radial basis function) to improve classification performance.
- Hyperparameters such as **C (regularization parameter) and gamma** are optimized.

Random Forest Classifier (RFC) Model

- RFC is an **ensemble learning method** that builds multiple decision trees and combines their results.
- The model randomly selects subsets of features and samples to train each decision tree, reducing overfitting.
- The number of trees and their depth are fine-tuned for optimal results.

3. Model Evaluation

Once the models are trained, their performance is evaluated using various metrics:

- **Accuracy**: Measures the percentage of correctly classified discussions.
- **Precision**: Determines how many of the predicted security issues are actually correct.

- **Recall (Sensitivity)**: Measures how many actual security issues were detected.
- **F1-Score**: Balances precision and recall for a more reliable evaluation.
- **Confusion Matrix**: Visualizes model performance by showing true positives, true negatives, false positives, and false negatives.

The performance of all three models is compared, and the best-performing model is selected.

4. Model Prediction and Deployment

Prediction on New Data

- The trained model is used to analyse new developer discussions and detect potential IoT security issues.
- The text from new discussions is processed using the same preprocessing and feature extraction techniques before passing it to the trained model.
- The model classifies whether the discussion contains a security-related concern.

DATASET DESCRIPTION

The dataset used in the project consists of developer discussions related to IoT security. It contains textual conversations where developers discuss various security aspects, vulnerabilities, and best practices. Each discussion entry is labeled to indicate whether it contains a security issue or not.

Dataset Structure

The dataset primarily has two key columns:

1. **Discussion Content** – This column contains text-based discussions, including developer conversations, forum posts, and issue reports.
2. **Security Issue** – This column acts as a label, indicating whether the discussion highlights a security concern. It is a binary classification with two possible values:
   - **"Yes" (1)** if the discussion contains a security-related issue.
   - **"No" (0)** if the discussion does not mention any security problem.

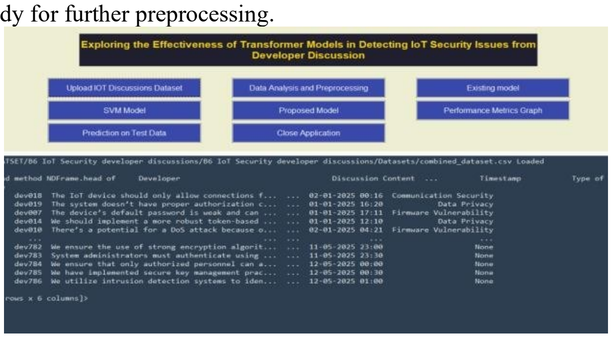
Data Preprocessing

Since the dataset consists of raw textual data, it needs several preprocessing steps before being fed into machine learning models:

1. **Text Cleaning** – Removing unnecessary symbols, special characters, and stopwords.
2. **Label Encoding** – Converting "Yes" and "No" into numerical values (1 and 0).
3. **Feature Extraction** – Transforming text into numerical representations using **TF-IDF (Term Frequency-Inverse Document Frequency)**.
4. **Splitting Data** – Dividing the dataset into training and testing sets for model evaluation.

RESULTS DESCRIPTION

In Fig. 2, the GUI displays the outcome immediately after a dataset has been uploaded. The file path or name is shown in the text console along with datatset (a preview) of the CSV dataset. This visual confirmation ensures that the correct file is loaded and that the data is ready for further preprocessing.


Exploring the Effectiveness of Transformer Models in Detecting IoT Security Issues from Developer Discussion
Upload IOT Discussions Dataset
Data Analysis and Preprocessing
Existing model
SVM Model
Proposed Model
Performance Metrics Graph
Prediction on Test Data
Close Application

```
TSET/86 IoT Security developer discussions/86 IoT Security developer discussions/Datasets/combined_dataset.csv Loaded

d method NDFrame.head of    Developer                               Discussion Content  ...        Timestamp          Type of

  dev018  The IoT device should only allow connections f...  ...  02-01-2025 00:16  Communication Security
  dev019  The system doesn't have proper authorization c...  ...  01-01-2025 16:20            Data Privacy
  dev007  The device's default password is weak and can ...  ...  01-01-2025 17:11  Firmware Vulnerability
  dev014  We should implement a more robust token-based ...  ...  01-01-2025 12:10            Data Privacy
  dev010  There's a potential for a DoS attack because o...  ...  02-01-2025 04:21  Firmware Vulnerability
     ...                                                ...  ...               ...                     ...
  dev782  We ensure the use of strong encryption algorit...  ...  11-05-2025 23:00                    None
  dev783  System administrators must authenticate using ...  ...  11-05-2025 23:30                    None
  dev784  We ensure that only authorized personnel can a...  ...  12-05-2025 00:00                    None
  dev785  We have implemented secure key management prac...  ...  12-05-2025 00:30                    None
  dev786  We utilize intrusion detection systems to iden...  ...  12-05-2025 01:00                    None

rows x 6 columns]>
```

Fig 2 GUI of detecting IOT security issues Uploading dataset.

In Fig. 3 The count plot in the project provides valuable insights into the distribution of different categories within the dataset.

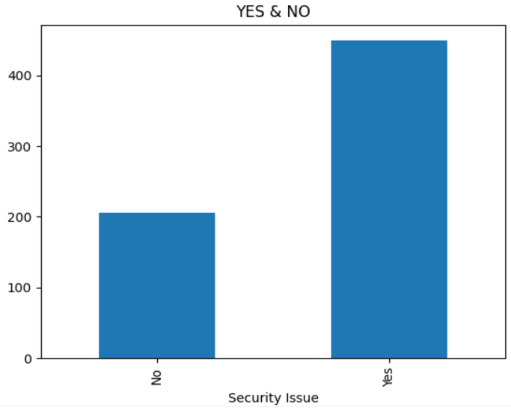
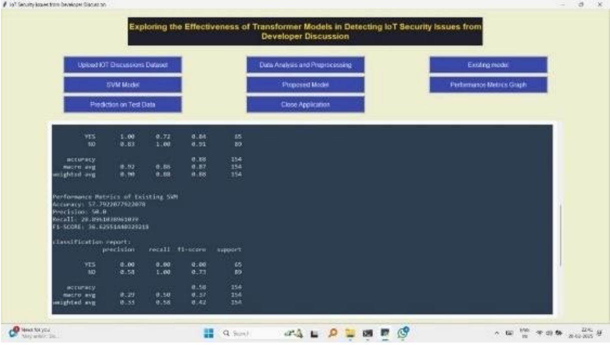


Fig 3 Count plot


IoT Security Issues from Developer Discussion
Exploring the Effectiveness of Transformer Models in Detecting IoT Security Issues from Developer Discussion
Upload IOT Discussions Dataset
Data Analysis and Preprocessing
Existing model
SVM Model
Proposed Model
Performance Metrics Graph
Prediction on Test Data
Close Application

```
         YES       1.00      0.72      0.84        65
          NO       0.83      1.00      0.91        89

    accuracy                           0.88       154
   macro avg       0.92      0.86      0.87       154
weighted avg       0.90      0.88      0.88       154

Performance Metrics of Existing SVM
Accuracy: 57.7922077922078
Precision: 50.0
Recall: 28.896103896103899
F1-SCORE: 36.62551440329218

classification report:
              precision    recall  f1-score   support

         YES       0.00      0.00      0.00        65
          NO       0.58      1.00      0.73        89

    accuracy                           0.58       154
   macro avg       0.29      0.50      0.37       154
weighted avg       0.33      0.58      0.42       154
```

Fig 4 GUI of proposed IOT security issues detection after applying model building and training using per XGBoost classifier.

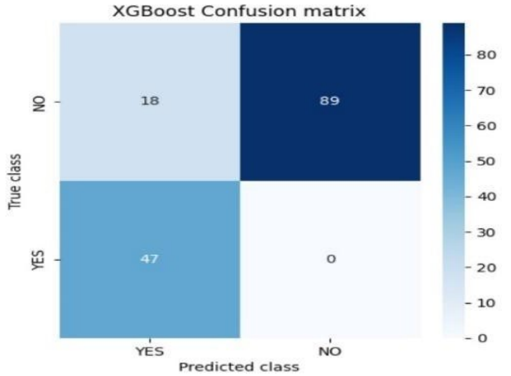
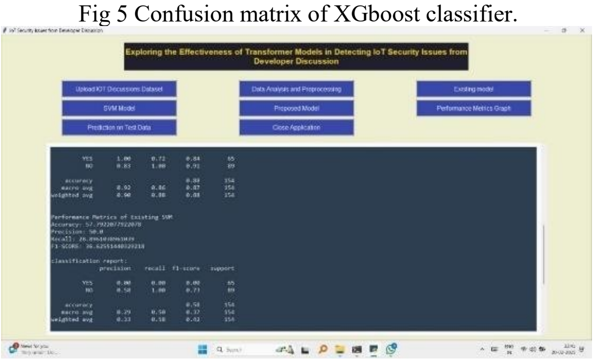


Fig 5 Confusion matrix of XGboost classifier.


IoT Security Issues from Developer Discussion
Exploring the Effectiveness of Transformer Models in Detecting IoT Security Issues from Developer Discussion
Upload IOT Discussions Dataset
Data Analysis and Preprocessing
Existing model
SVM Model
Proposed Model
Performance Metrics Graph
Prediction on Test Data
Close Application

```
         YES       1.00      0.72      0.84        65
          NO       0.83      1.00      0.91        89

    accuracy                           0.88       154
   macro avg       0.92      0.86      0.87       154
weighted avg       0.90      0.88      0.88       154

Performance Metrics of Existing SVM
Accuracy: 57.7922077922078
Precision: 50.0
Recall: 28.896103896103899
F1-SCORE: 36.62551440329218

classification report:
              precision    recall  f1-score   support

         YES       0.00      0.00      0.00        65
          NO       0.58      1.00      0.73        89

    accuracy                           0.58       154
   macro avg       0.29      0.50      0.37       154
weighted avg       0.33      0.58      0.42       154
```

Fig 6 GUI of proposed IOT security issues detection using SVM model.

The confusion matrix generated for the SVM classifier is shown in Fig. 7. The matrix is typically rendered as a heatmap where rows represent the actual classes and columns represent the predicted classes.

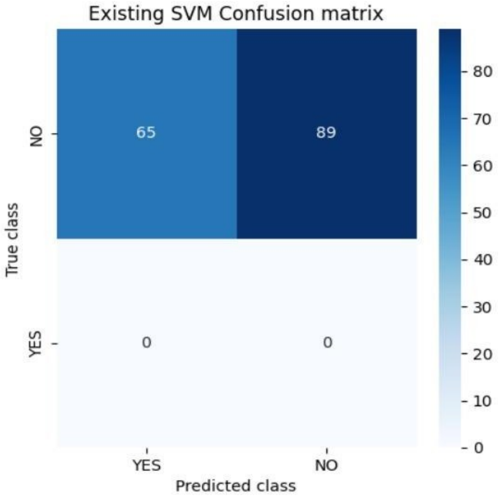
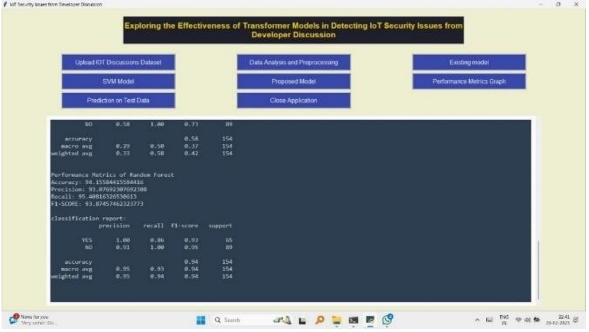


Fig 7 Confusion matrix obtained using SVM classifier.

In Fig. 8, the GUI is updated to reflect the results of training Random Forest Classifier. Similar to the XGboost abn SVM training GUI.


IoT Security Issues from Developer Discussion
Exploring the Effectiveness of Transformer Models in Detecting IoT Security Issues from Developer Discussion
Upload IOT Discussions Dataset
Data Analysis and Preprocessing
Existing model
SVM Model
Proposed Model
Performance Metrics Graph
Prediction on Test Data
Close Application

```
          NO       0.58      1.00      0.73        89

    accuracy                           0.58       154
   macro avg       0.29      0.50      0.37       154
weighted avg       0.33      0.58      0.42       154


Performance Metrics of Random Forest
Accuracy: 94.15584415584416
Precision: 93.07692307692308
Recall: 95.40816326530613
F1-SCORE: 93.87457462323773

classification report:
              precision    recall  f1-score   support

         YES       1.00      0.86      0.93        65
          NO       0.91      1.00      0.95        89

    accuracy                           0.94       154
   macro avg       0.95      0.93      0.94       154
weighted avg       0.95      0.94      0.94       154
```

Fig 8 GUI of proposed detection pf IOT security issues using RFC model.

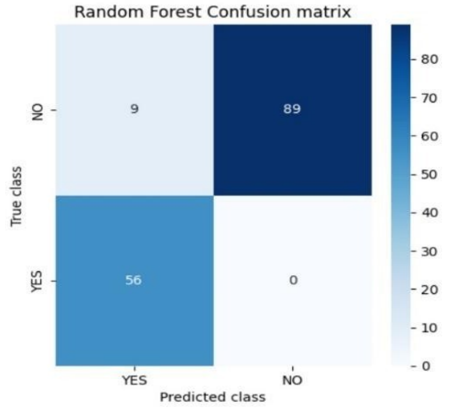


Fig 9 Confusion matrix using RFC model

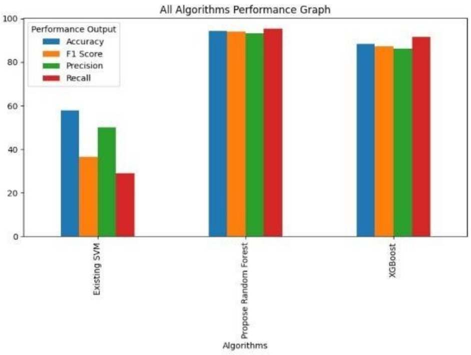
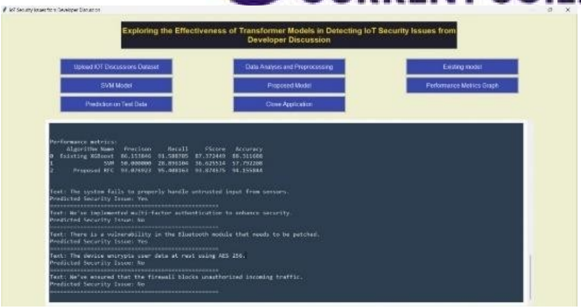


Fig. 10: Performance comparison graph of XGBoost , RFC models.


IoT Security Issues from Developer Discussion
Exploring the Effectiveness of Transformer Models in Detecting IoT Security Issues from Developer Discussion
Upload IOT Discussions Dataset
Data Analysis and Preprocessing
Existing model
SVM Model
Proposed Model
Performance Metrics Graph
Prediction on Test Data
Close Application

```
Performance metrics:
     Algorithm Name   Precison     Recall     FScore   Accuracy
0  Existing XGBoost  86.153846  91.588785  87.372449  88.311688
1               SVM  50.000000  28.896104  36.625514  57.792208
2      Proposed RFC  93.076923  95.408163  93.874575  94.155844


Text: The system fails to properly handle untrusted input from sensors.
Predicted Security Issue: Yes
==================================================
Text: We've implemented multi-factor authentication to enhance security.
Predicted Security Issue: No
==================================================
Text: There is a vulnerability in the Bluetooth module that needs to be patched.
Predicted Security Issue: Yes
==================================================
Text: The device encrypts user data at rest using AES 256.
Predicted Security Issue: No
==================================================
Text: We've ensured that the firewall blocks unauthorized incoming traffic.
Predicted Security Issue: No
==================================================
```

Fig 11: sample predictions on test data using proposed RFC model.

Algorithm Name	Accuracy	Precision	Recall	f1-score
XGboost Classifier	86.15%	91.58%	87.37%	88.31%
SVM Classifier	50.00%	28.89%	36.62%	57.79%
RFC Classifier	93.07	95.40	93.87	94.15

Table 1: Summarizing the performance metrics for the two models.

5. CONCLUSION

The research demonstrates the effectiveness of transformer-based models in detecting IoT security issues from developer discussions. By analysing textual data, the system can classify security-related concerns with high accuracy, helping organizations and developers proactively identify potential threats. Traditional models such as SVM, Random Forest, and XGBoost were used for comparison, showing the benefits and limitations of each approach. The results indicate that AI-driven security issue detection enhances automated threat monitoring and assists in mitigating risks before they escalate. Overall, this project contributes to advancing cybersecurity measures in IoT environments by leveraging machine learning for intelligent security assessment.

Another way to look into IoT issues is to look at IoT developer discussions on major online development forums like Stack Overflow (SO). However, finding discussions that are relevant to IoT issues is challenging since they are frequently not categorized with IoT-related terms. In this paper, we present the "IoT Security Dataset", a domain-specific dataset of 7147 samples focused solely on IoT security discussions. As there are no automated tools to label these samples, we manually labeled them. We further employed multiple transformer models to automatically detect security discussions. Through rigorous investigations, we found that IoT security discussions are different and more complex than traditional security discussions. We demonstrated a considerable performance loss (up to 44%) of transformer models on cross-domain datasets when we transferred knowledge from a general-purpose dataset "Opiner", supporting our claim. Thus, we built a domain-specific IoT security detector with an F1-Score of 0.69. We have made the dataset public in the hope that developers would learn more about the security discussion and vendors would enhance their concerns about product security.

REFERENCES

[1]. DevlinJacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre- training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

[2]. Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692.

[3]. Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems.

[4]. Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. "Code and named entity recognition in stackoverflow." arXiv preprint arXiv:2005.01634.

[5]. Uddin, Gias, Fatima Sabir, Yann-Gaël Guéhéneuc, Omar Alam, and Foutse Khomh. "An empirical study of IoT topics in IoT developer discussions on Stack Overflow." Empirical Software Engineering 26, no. 6.

[6]. Ahmed, S.W.; Kientz, F.; Kashef, R. A modified transformer neural network (MTNN) for robust intrusion detection in IoT networks. In Proceedings of the 2023 International Telecommunications Conference (ITC-Egypt), Alexandria, Egypt, 18–20 July 2023; pp. 663–668.

[7]. He, M.S.; Wang, X.J.; Wei, P.; Yang, L.; Teng, Y.L.; Lyu, R.J. Reinforcement learning meets network intrusion detection: A transferable and adaptable framework for anomaly behavior identification. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 2477–2492

[8]. Jony, A.I.; Arnob, A.K.B. A long short-term memory-based approach for detecting cyber-attacks in IoT using CIC-IoT2023 dataset. *J. Edge Comput.* **2024**, *3*, 28–42.

[9]. Jaradat, A.S.; Nasayreh, A.; Al-Na'amneh, Q.; Gharaibeh, H.; Al Mamlook, R.E. Genetic optimization techniques for enhancing web attacks classification in machine learning. In Proceedings of the IEEE International Conference on 11 Dependable 2023, Autonomic & Secure Computing, Abu Dhabi, United Arab Emirates, 14–17 November 2023; pp. 0130–0136.

[10]. Abbas, S.; Al Hejaili, A.; Sampedro, G.A.; Abisado, M.A.; Almadhor, A.M.; Shahzad, T.; Ouahada, K. A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures. *IEEE Access* **2023**, *11*, 112189–112198.