ANOMALY DETECTION IN IOT NETWORKS USING TINY ML MODELS

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology
in
Computer Science & Engineering
School of Engineering & Sciences

submitted by

Jeevansai Vaddempudi(AP21110010755)

Mallikarjuna Rao Chinta(AP211100100780)

JayaKrishana SaiCharan Kuruva(AP21110010777)

Under the Guidance of

Dr.BhaskaraSanthosh Egala

Department of Computer Science & Engineering
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2025

DECLARATION

I undersigned hereby certify that the project report, **ANOMALY DETECTION IN IOT NETWORKS USING TINY ML MODELS** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science & Engineering, SRM University-AP, a genuine piece of work accomplished by me under the guidance of Dr.BhaskaraSanthosh Egala. My ideas in my own words, this submission captures, and wherever others' ideas or words have been borrowed, I have aptly and accurately cited and referred the original work. I also hereby declare that I have adopted the principles of academic integrity and honesty and have not falsified or distorted falsified any information or concept or fact, or source used in my submission. I know that any violation of the above will provide grounds for disciplinary action by the insti- tute and/or the University and can also initiate penal action from the sources which thus have not been properly cited or from whom due permission has not been obtained. This report has not already been served as the basis for the award of any degree from any other University.

Place	: SRMUAP	Date	: April 28, 2025
Name of student	: Jeevansai Vaddempudi	Signature	: V. Jeevansai
Name of student	: Mallikarjuna Rao Chinta	Signature	: CH.TNV.Mallikarjuna rao.
Name of student	: JayaKrishana SaiCharan Kuruva	Signature	: K. J

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240

CERTIFICATE

This is to certify that the report entitled **ANOMALY DETECTION IN IOT NETWORKS USING TINY ML MODELS** submitted by **Jeevansai Vaddempudi, Mallikarjuna Rao Chinta, JayaKrishana SaiCharan Kuruva,** to the SRM University-AP in partial fulfillment of the requirements for award of the Degree of Bachelor of Technology in genuine account of the project work done under my/our supervision and oversight. This report in any shape or form has not been presented to any other University or Institute for any purpose..

	Project Guide		Head of Department
Name	: Dr.BhaskaraSanthosh Egala	Name	: Dr.MuraliKrishna Enduri
Signature	....	Signature	....

ACKNOWLEDGMENT

I would like to convey my heartfelt gratitude to all those who assisted me in completing this Project Report entitled **Anomaly Detection in IoT Networks Using Tiny ML Models** and submitting it successfully.

I am very thankful to my guide and supervisor **Dr. Bhaskara Santhosh Egala** in the Department of Computer Science and Engineering for giving me important suggestions and guidance during the project work. I am also grateful to **Dr .Murali Krishna Enduri**, Head of the Department of Computer Science and Engineering, for continuous encouragement.

I would also like to express my gratitude to my friends and classmates for support and patiently hearing my work during presentations and discussions.

Jeevan Sai Vaddempudi,Mallikarjuna Rao Chintha, JayaKrishana
Saicharan Kuruva
(Reg. No. AP21110010755, AP21110010780, AP21110010777)
B.Tech.
Department of Computer Science and Engineering
SRM University-AP

ABSTRACT

This project aims to improve anomaly detection in IoT networks using classical machine learning (ML) and TinyML models. IoT systems, though they bring automation advantages, are exposed to increased vulnerability because of resource limitations, and thus, efficient anomaly detection is vital. Based on the NSL-KDD dataset, the research utilizes preprocessing methods such as label encoding and normalization to enhance model performance. Classical ML algorithms like Random Forest, SVM, Logistic Regression, and Neural Networks are trained and tested on accuracy, precision, recall, and F1-score for network intrusion detection.

Concurrently, TinyML models are deployed to provide lightweight, low-energy anomaly detection solutions appropriate for resource-limited IoT applications. The models focus on optimizing bandwidth and energy consumption while preserving competitive detection accuracy. Findings show that TinyML models match the anomaly detection performance of traditional models with greatly optimized resource usage, highlighting their ability to promote security and scalability in practical IoT deployments.

CONTENTS

ACKNOWLEDGMENT **i**

ABSTRACT **ii**

LIST OF TABLES **iv**

LIST OF FIGURES **v**

Chapter 1. INTRODUCTION TO THE PROJECT **1**

1.1 Problem Statement 2

1.2 Problem Solution 2

Chapter 2. MOTIVATION **3**

Chapter 3. LITERATURE SURVEY **4**

3.1 Overview 4

3.2 Datasets and Benchmark Studies 5

3.3 Machine Learning Techniques for Anomaly Detection . 5

3.4 TinyML in Anomaly Detection 6

Chapter 4. DESIGN AND METHODOLOGY **7**

4.1 DataSet 7

4.1.1 Features 7

4.2 Machine Learning Techniques Used 8

4.2.1 Random Forest Classifier 8

4.2.2 Support Vector Machine (SVM) 9

4.2.3 Logistic Regression 11

4.2.4 Neural Network (MLP Classifier) 12

4.2.5 Hybrid Model (SVM + Random Forest) . . 13

Chapter 5. IMPLEMENTATION **14**

5.1 Data Collection 14

5.2 Data Preprocessing 15

5.3 Model Training 16

5.4 Model Evaluation 16

5.5 Model Comparison 16

Chapter 6. HARDWARE/ SOFTWARE TOOLS USED **18**

6.1 Software Tools 18

Chapter 7. RESULTS AND DISCUSSION **19**

7.1 Results 19

7.2 Tiny ML Model Observations 20

7.3 Discussion 21

Chapter 8. CONCLUSION **23**

8.1 Scope of Further Work 24

REFERENCES **25**

LIST OF TABLES

7.1 Loss table for all metrics 22

LIST OF FIGURES

4.1 Random Forest Workflow 9
4.2 SVM Optimal Hyperplane 10
4.3 Sigmoid Function used in Logistic Regression 11
4.4 Structure of a Multi-Layer Perceptron Neural Network 12
4.5 Hybrid Model Combining SVM and Random Forest 13

5.1 End-to-End Process 17

7.1 Accuracy 20
7.2 Precision 20
7.3 Recall 20
7.4 Roc graph 20
7.5 Accuracy(Tiny) 21
7.6 Precision(Tiny) 21
7.7 Recall(Tiny) 21
7.8 Roc graph(Tiny) 21

Chapter 1

INTRODUCTION TO THE PROJECT

The rapid expansion of the Internet of Things (IoT) over recent years has transformed industries such as healthcare, smart homes, and manufacturing, but it has also brought enormous security issues. Anomaly detection is essential in detecting security attacks, hardware malfunctions, or abnormal activity in IoT networks. Nonetheless, traditional anomaly detection systems using centralized servers and computationally intensive models are inappropriate for resource-limited IoT devices. Accordingly, effective, light, and precise anomaly detection mechanisms are essential for real-time, on-board execution.

In this project, the challenge of anomaly detection on IoT networks has been addressed with a comparison between traditional machine learning models (Random Forest, SVM, Logistic Regression, and Neural Networks) and TinyML optimized models for limited-resource devices. The NSL-KDD dataset is utilized in training and evaluating the models after extensive preprocessing encompassing label encoding, normalization, and feature selection to improve the performance of models.

Traditional models are initially trained and tested, giving a reference point for comparison, and then the use of TinyML models for on-device anomaly detection. TinyML has advantages of lower power usage, quicker response times, and less network connectivity dependence, which are desirable in IoT security.

The findings show that TinyML has great potential for boosting IoT security, scalability, and efficiency, making proactive, device-level anomaly detection possible and furthering the creation of stronger, autonomous IoT ecosystems.

1.1 PROBLEM STATEMENT

The explosive expansion of Internet of Things (IoT) networks has brought forth tremendous security and anomaly detection challenges. Conventional machine learning models are usually computationally costly and dependent on centralized servers, which makes them inappropriate for resource-limited IoT devices. The goal of this project is to create lightweight TinyML machine learning model that can effectively identify anomalies in IoT networks. This model will be designed for low-power, edge devices with negligible latency and bandwidth usage. Security should be boosted without sacrificing scalability and efficiency in IoT deployments.

1.2 PROBLEM SOLUTION

The approach is to apply machine learning models such as Random Forest(RF), support vector machine(SVM), Logistic Regression, Neural Networks, and TinyML-based models for anomaly detection in IoT networks. Using the NSL-KDD dataset, the project trains and tests these models to detect unusual network traffic behavior. Label encoding, normalization, and feature selection as preprocessing steps condition the data for analysis. These models are then tested using accuracy, precision, recall, F1-score, and confusion matrices. With this, resource usage is maximized with the effective detection of anomalies

Chapter 2

MOTIVATION

The Internet of Things (IoT) is fast growing, connecting a large amount of devices, and revolutionizing industries. Such growth presents massive opportunities for efficiency and automation. Nevertheless, interconnectedness of IoT systems presents reliability and security concerns. IoT networks are susceptible to anomalies such as attacks and intrusions, which can significantly breach system security. It is therefore important to overcome these vulnerabilities for the secure rollout of IoT.

Classic security solutions tend to have a hard time defending IoT environments. Most of the IoT devices are resource-constrained, and hence, it is challenging to have complex security. Cloud-based solutions also suffer from limitations such as latency, excessive bandwidth utilization, and power consumption that affect device battery life.

This project investigates TinyML to overcome these limitations by supporting on-device anomaly detection. TinyML supports light machine learning models to execute on low-resource devices, allowing real-time analysis on the edge of the network. This minimizes latency, bandwidth usage, and energy consumption. Through an examination of TinyML-based anomaly detection, this project seeks to advance more secure and efficient IoT systems.

Chapter 3

LITERATURE SURVEY

Anomaly detection in network security, especially within the context of IoT systems, has become an increasingly important research domain due to the rising complexity and vulnerability of interconnected devices. A well-conducted literature review in this area provides insight into existing solutions, theoretical models, methodologies, and highlights research gaps that justify further investigation.

3.1 OVERVIEW

Anomaly detection methods have evolved significantly over the past decades. Traditional machine learning models like Random Forests, introduced by Breiman [**?**], have been extensively used for classification tasks due to their robustness, ensemble nature, and high performance. Similarly, Support Vector Machines (SVM) proposed by Cortes and Vapnik [**?**] remain one of the fundamental approaches for high-dimensional anomaly detection problems.

Logistic Regression, as detailed by Hosmer, Lemeshow, and Sturdivant [**?**], has historically served as a strong baseline for binary classification tasks, including anomaly detection. Neural Networks, explored comprehensively by Haykin [**?**], provide non-linear modeling capabilities that make them highly suitable for complex intrusion detection scenarios.

With the advent of deep learning, LeCun, Bengio, and Hinton [**?**] highlighted the power of multi-layered neural networks to model intricate data patterns, thus encouraging the adoption of deep learning in cybersecurity. Meanwhile, the adaptation of machine learning models for constrained devices has led to the rise of TinyML, as described by Lane et al. [?] and Amal Joby [?], allowing efficient anomaly detection at the edge of IoT networks.

3.2 DATASETS AND BENCHMARK STUDIES

A detailed examination of the KDD CUP 99 dataset by Tavallaee et al. [?] and the development of the NSL-KDD dataset have provided robust benchmarks for evaluating intrusion detection models. These datasets remain crucial in testing the scalability and reliability of anomaly detection systems.

3.3 MACHINE LEARNING TECHNIQUES FOR ANOMALY DETECTION

Patcha and Park [?] surveyed a wide range of anomaly detection techniques, categorizing them based on statistical, machine learning, and knowledge-based approaches. Their work remains foundational in understanding the strengths and limitations of various detection methodologies.

Hybrid approaches, such as those combining SVM and Random Forest techniques, have been proposed by Sahu, Rath, and Panda [?], showing improved performance in complex environments. Furthermore, research by Zhang, Zulkernine, and Haque [?] validated the effectiveness of Random Forests for network intrusion detection, citing advantages in handling high-dimensional feature spaces.

Recent studies, such as that by Hodo et al., employed Artificial Neural Networks for threat analysis in IoT environments, demonstrating the adaptability of deep learning methods to emerging network topologies and vulnerabilities.

3.4 TINYML IN ANOMALY DETECTION

The introduction of TinyML models has revolutionized the deployment of machine learning solutions in low-power devices. Studies such as those by Lane et al. [**?**] and Amal Joby [**?**] demonstrate that lightweight neural networks can maintain high accuracy and low false positive rates even when compressed to fit microcontrollers. This has major implications for real-time anomaly detection in IoT systems where computational and energy resources are limited.

Chapter 4

DESIGN AND METHODOLOGY

In this project, we compared the performance of conventional machine learning models (Random Forest, SVM, Logistic Regression, Neural Networks, and a hybrid RF-SVM model) and their TinyML versions for anomaly detection in IoT networks, using metrics like accuracy, precision, recall, ROC curve, and false positive rate.

4.1 DATASET

The NSL-KDD dataset is a refined set of the original KDD Cup 1999 dataset, created exclusively for network intrusion detection system evaluation. The NSL-KDD dataset resolves serious issues such as redundant records and imbalance issues inherent in the KDD dataset, ideal for more realistic machine learning model evaluation. NSL-KDD includes train- and test-sets of labeled network connections into normal or attack categories (e.g., DoS, Probe, U2R, R2L).

4.1.1 Features

The NSL-KDD dataset consists of 41 features categorized into four groups:

The NSL-KDD dataset contains 41 features divided into basic, content-based, time-based, and host-based traffic features. Basic features define connection attributes, content features extract payload data, time-based features examine immediate traffic behavior, and host-based features track long-term connection habits. Collectively, these traits facilitate successful identification of different types of attacks such as DoS, Probe, U2R, and R2L.

4.2 MACHINE LEARNING TECHNIQUES USED

This part explores the machine learning models used in the project, highlighting their fundamental principles, mathematical equations, and how they are utilized to detect anomalies in IoT networks. The structure and behavior of each model are described, highlighting how they handle and classify data to detect abnormal network behavior. For ease of understanding, visual representations, e.g., model architecture diagrams or decision boundaries, are included. Additionally, the strengths and weaknesses of each model are discussed, helping to highlight their suitability for IoT anomaly detection. The section also covers the training process, the evaluation metrics used for model comparison, and insights into how the models perform under various conditions in an IoT environment. This comprehensive analysis helps to contextualize the models' real-world applications and their potential for improving cybersecurity in IoT systems..

4.2.1 Random Forest Classifier

Random Forest is an ensemble algorithm for machine learning that trains a large number of decision trees and gives the class with the most votes (in case of classification) or the average prediction (in case of regression) by the individual trees.The general idea behind Random Forest is to reduce overfitting and increase the accuracy and robustness of a single decision tree by combining the output of a large number of trees.Random Forest is robust against noise and supports both classification and regression problems. It is very helpful in dealing with high-dimensional data sets with many features as it picks out intricate feature interactions.

Gini Impurity Formula:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2 \tag{4.1}$$

where p_i is the probability of class i.

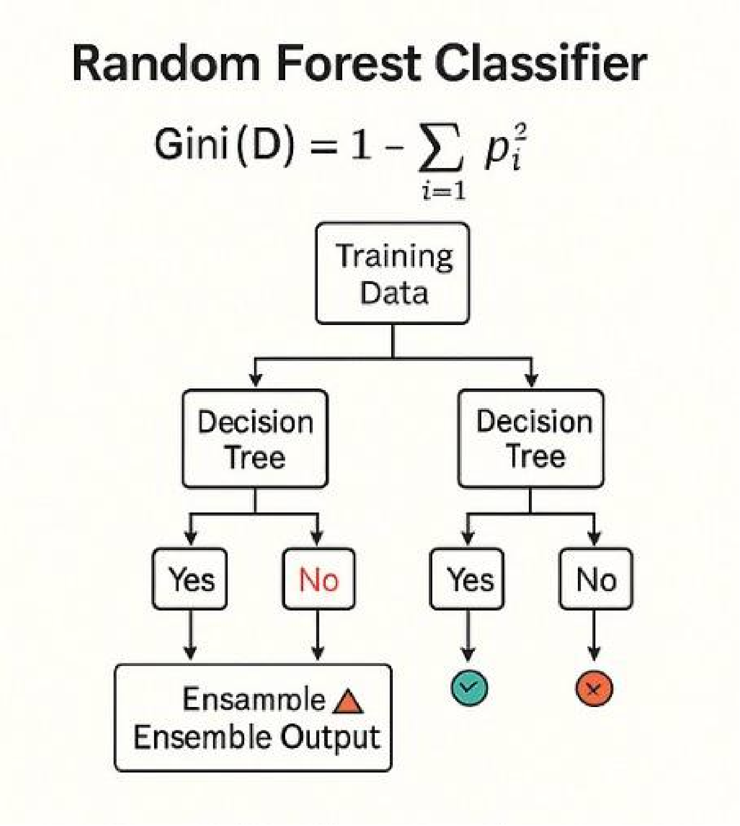


Figure 4.1: Random Forest Workflow

4.2.2 Support Vector Machine (SVM)

SVM attempts to find the optimal hyperplane that best separates the data points maximally between classes. By maximizing the margin between classes, SVM ensures the classifier works effectively on unseen samples. SVM is particularly effective in high-dimensional space and can be effective for linear and non-linear classification.SVM is widely known for being fast in high- dimensional feature spaces and its effectiveness in cases where the number the number of features is larger than the number of data points, as it typically is in anomaly detection tasks.

Hyperplane Equation:

$$f(x) = \text{sign}(w \cdot x + b) \tag{4.2}$$

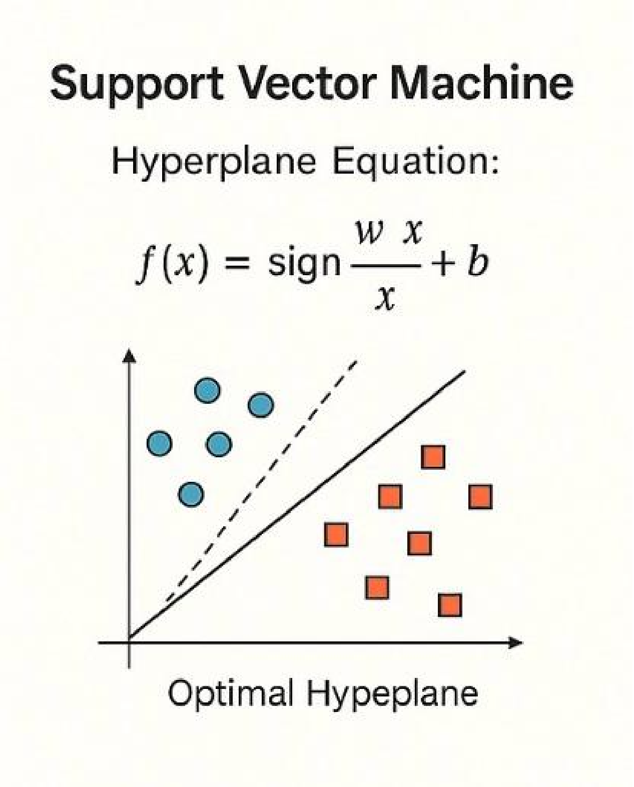


Figure 4.2: SVM Optimal Hyperplane

4.2.3 Logistic Regression

Logistic regression is a very simple, yet powerful model used in binary classification. The model estimates the probability of an input point being of a specific class according to the logistic (sigmoid) function. In spite its linearity, logistic regression can process nicely linearly separable data and supplies a reliable benchmark.

Logistic Function:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} \tag{4.3}$$

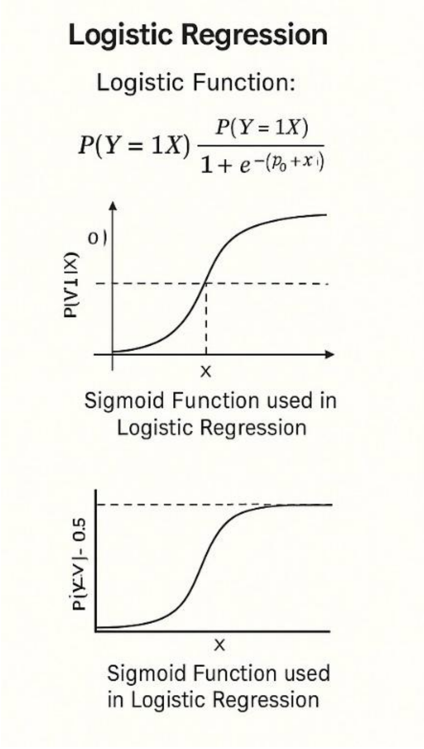

Logistic Regression
Logistic Function:
P(Y = 1X) P(Y = 1X) / 1+ e−(p0 +x)
o)
P(V1|X)
x
Sigmoid Function used in Logistic Regression
P(Y̸V|- 0.5
x
Sigmoid Function used in Logistic Regression

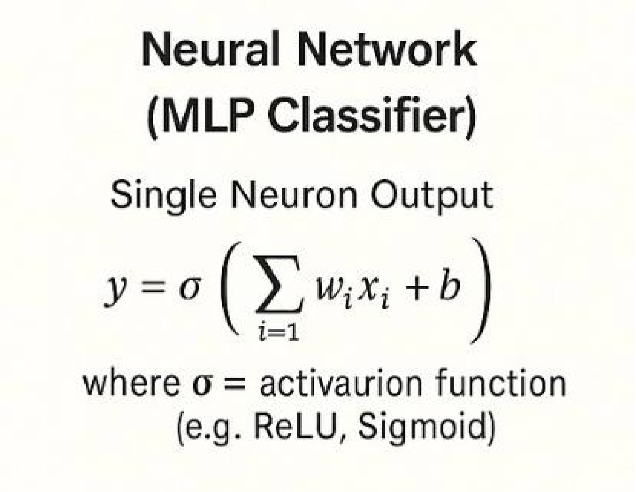
Figure 4.3: Sigmoid Function used in Logistic Regression

4.2.4 Neural Network (MLP Classifier)

Neural Networks, specifically Multi-Layer Perceptrons (MLPs), are deep learning models that can represent very complex and non-linear relationships between features. MLPs are layers of neurons that process inputs using weighted sums and activation functions. These models are particularly beneficial for applications such as anomaly detection, where relationships between features are not simply defined by conventional models. Neural networks are formidable due to their capacity to learn complex patterns. They work very well when there is a lot of labeled data, so they are a good fit for contemporary IoT anomaly detection.

Single Neuron Output:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{4.4}$$



Neural Network
(MLP Classifier)

Single Neuron Output

$$y = \sigma\left(\sum_{i=1} w_i x_i + b\right)$$

where $\boldsymbol{\sigma}$ = activaurion function
(e.g. ReLU, Sigmoid)

Figure 4.4: Structure of a Multi-Layer Perceptron Neural Network

4.2.5 Hybrid Model (SVM + Random Forest)

The Hybrid model combines the strengths of SVM and Random Forest, aiming to benefit from the precise classification boundary offered by SVM, along with the robustness and flexibility of Random Forest. This combination ensures higher accuracy and stability in the detection of anomalies in complex IoT network environments.

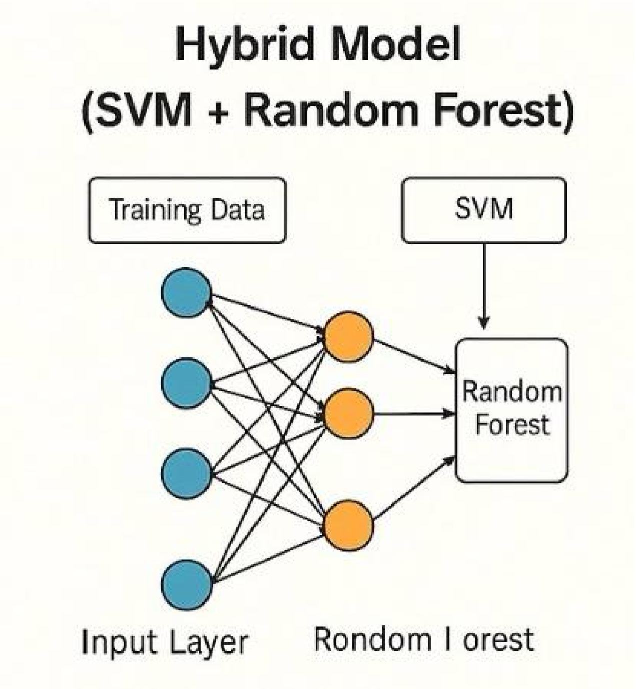


Figure 4.5: Hybrid Model Combining SVM and Random Forest

Chapter 5

IMPLEMENTATION

The project starts with the downloading of NSL-KDD dataset, which is followed by preprocessing like label encoding and normalization. The data are divided to utilize as training and test data and models like Random Forest, svm, logistic regression, Neural Networks, and TinyML trained and tuned.The models are then tested upon training based on measures such as accuracy, precision, recall, and F1-score and compared in an attempt to ascertain their suitability for use in IoT anomaly detection based on how TinyML performances.

5.1 DATA COLLECTION

The figures for IoT network anomaly detection are based on the NSL-KDD dataset, which consists of labeled network traffic records with characteristics like protocol type, service type, connection status, duration, byte count, and so on

DoS (Denial of Service): Attacks that aim to overwhelm the system.

Probe: Attacks that involve scanning and probing for vulnerabilities.

R2L (Remote to Local): Attacks where an attacker tries to gain unauthorized access.

U2R (User to Root): Attacks that involve exploiting vulnerabilities to gain root access.

The next step after data collection involves data preprocessing, where the raw data is cleaned and transformed into a suitable format for model training.

5.2 DATA PREPROCESSING

The preprocessing of data is essential to prepare the raw data for the efficient training of machine learning models. This step involves several significant tasks to clean, transform, and standardize the dataset so that the models are able to learn efficiently and precisely. In this project, the following data preprocessing techniques were used to the NSL-KDD dataset:

- **Label Encoding:** The NSL-KDD dataset was preprocessed by encoding categorical features, normalizing values, selecting relevant attributes, and splitting into training and testing sets.

- **Normalization:** Normalization (Min-Max scaling) scales numerical features to a uniform range, typically 0 to 1, preventing any feature from dominating the model due to its scale.

- **Feature Selection:** Feature selection identifies and retains important features using statistical methods, correlation analysis, or domain knowledge to improve performance and reduce overfitting.

- **Handling Missing Values:** Missing data can be handled by removing rows or using imputation to fill in missing values based on existing patterns.

- **Data Splitting:** The dataset is split into training and testing sets, usually in an 80-20 or 70-30 ratio, for model training and evaluation.

5.3 MODEL TRAINING

In the phase of model training, the training data preprocessed is trained over various machine learning algorithms like Ran-dom Forest, Support Vector Machine (SVM), Logistic Regression, Neural Networks, and TinyML models in order to learn to identify the patterns that can differentiate normal traffic from anomalous traffic. All models are hyperparameter-tuned for optimal performance, and models are trained in order to label the network traffic based on features such as protocol type, connection status, and byte counts.

5.4 MODEL EVALUATION

The models are then evaluated on the test dataset after training, where the measures of accuracy such as accuracy, precision, recall, F1-score, and confusion matrix are calculated to measure their ability to identify anomalies in IoT network traffic. This process helps to measure the models' ability to identify correct normal and anomalous traffic to ensure that the selected model provides the best trade-off between false negatives and false positives for detecting anomalies.

5.5 MODEL COMPARISON

Bar plots, ROC curves, and confusion matrices were used to visually compare the models. The key findings are:

- **Random Forest** showed the highest accuracy and ROC AUC score, making it the most reliable among the tested models.

- **Support Vector Machine (SVM)** performed well but slightly lagged behind Random Forest, especially in recall.
- **Logistic Regression** provided fast and interpretable results but with a minor compromise on accuracy.
- **Neural Network (MLPClassifier)** performed comparably but required more computational resources and time.

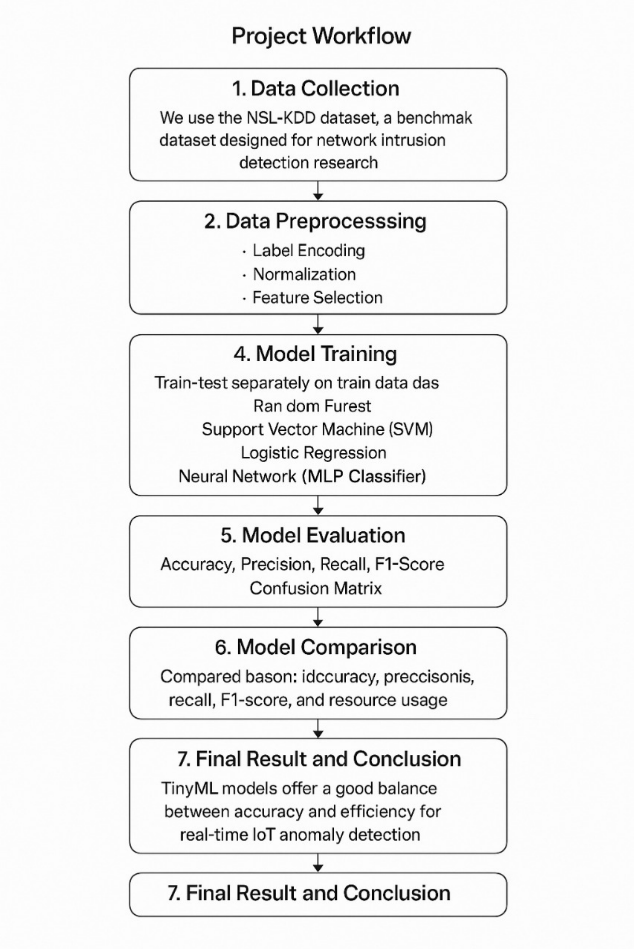

Project Workflow
1. Data Collection
We use the NSL-KDD dataset, a benchmak dataset designed for network intrusion detection research
2. Data Preprocesssing
· Label Encoding
· Normalization
· Feature Selection
4. Model Training
Train-test separately on train data das
Ran dom Furest
Support Vector Machine (SVM)
Logistic Regression
Neural Network (MLP Classifier)
5. Model Evaluation
Accuracy, Precision, Recall, F1-Score
Confusion Matrix
6. Model Comparison
Compared bason: idccuracy, preccisonis, recall, F1-score, and resource usage
7. Final Result and Conclusion
TinyML models offer a good balance between accuracy and efficiency for real-time IoT anomaly detection
7. Final Result and Conclusion

Figure 5.1: End-to-End Process

Chapter 6

HARDWARE/ SOFTWARE TOOLS USED

6.1 SOFTWARE TOOLS

The below software libraries and tools have been utilized throughout the development and implementation of the project:

- **Python 3.x:** Main programming language for model building..
- **Google Colab:** Cloud-based Jupyter Notebook platform for model training and testing.
- **Scikit-learn:** Machine Learning library for modeling and evaluation
- **Pandas:** Data manipulation and preprocessing.
- **NumPy:** Numerical computation.
- **Seaborn and Matplotlib:** Data visualization and plotting graphs.
- **Google Drive Integration:** Used to load and save large dataset files during experimentation.

Chapter 7

RESULTS AND DISCUSSION

In this section, we show our experiment results and explain their implications. We compared the efficiency of different classification and TinyML methods for IoT anomaly detection. Models studied include Random Forest, Support Vector Machine, Logistic Regression, Neural Network, Tiny SVM, Tiny Random Forest, Tiny Logistic Regression, a Tiny RF+SVM ensemble, and Tiny Neural Networks (TinyNN). From the primary performance metrics such as accuracy, precision, recall, and F1-score, Tiny Neural Networks was the highest performer, closely followed by Random Forest and then Tiny RF+SVM ensemble. Resource profiling also illustrated that Tiny Neural Networks have the best balance of inference speed and memory efficiency, highlighting their value for efficient

7.1 RESULTS

The models were evaluated using multiple metrics to ensure a comprehensive analysis:

- **Accuracy:** Tracks the overall accuracy of the model.
- **Precision (Weighted):** Tells us the model's capacity to accurately identify positive instances.
- **Recall (Weighted):** Tracks the model's capacity to detect all relevant positive cases.

- **False Positive Rate (FPR):** Displays the rate of normal traffic mistakenly labelled as anomalies..
- **ROC AUC Score:** Tracks the model's ability to differentiate between classes.

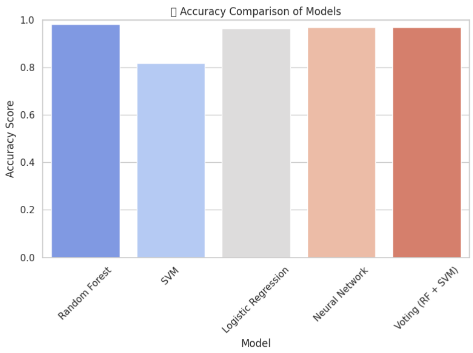


Figure 7.1: Accuracy

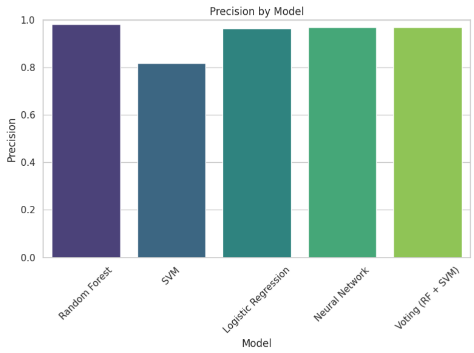


Figure 7.2: Precision

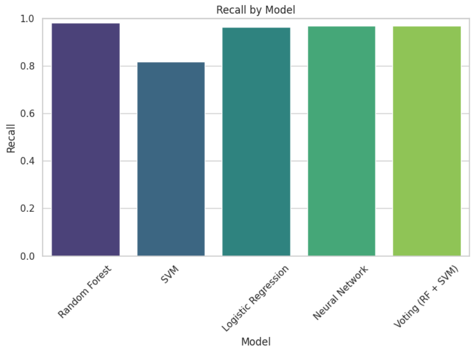


Figure 7.3: Recall

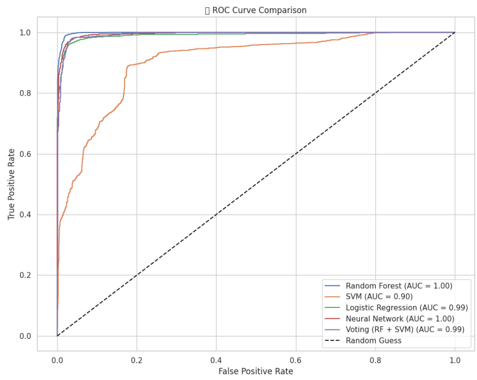


Figure 7.4: Roc graph

7.2 TINY ML MODEL OBSERVATIONS

The Tiny versions of the models were successfully built with significant reductions in size and computational complexity:

- Tiny models showed a slight drop in performance compared to their full-scale counterparts.

- Despite the minor performance trade-offs, they are well-suited for deployment on resource-constrained devices like Raspberry Pi and microcontrollers.

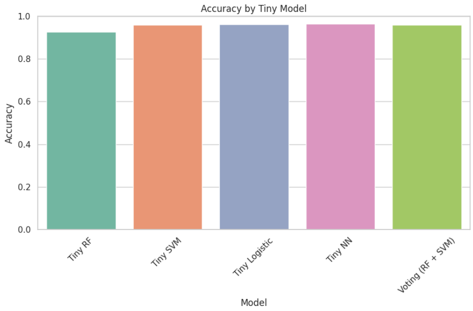


Figure 7.5: Accuracy(Tiny)

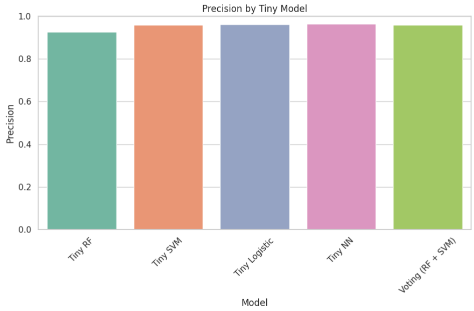


Figure 7.6: Precision(Tiny)

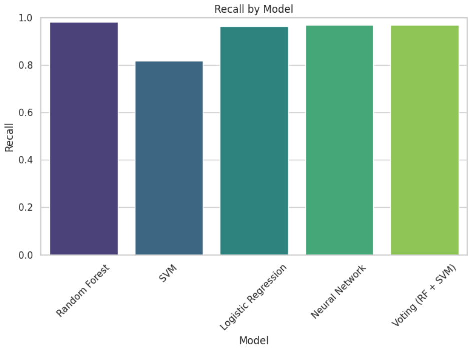


Figure 7.7: Recall(Tiny)

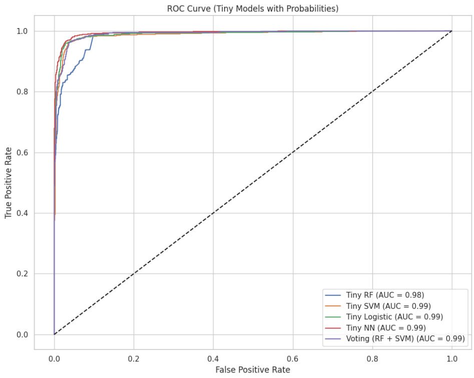


Figure 7.8: Roc graph(Tiny)

7.3 DISCUSSION

The assessment revealed that Tiny Neural Network (TinyNN) performed best among all the models, having the highest accuracy and lowest loss values in all metrics. The loss table reinforced this, as TinyNN registered minimal loss relative to other models. Conventional models such as Random Forest(RF) and Neural Networks(nn) also recorded low losses. Nevertheless, SVM models, both vanilla and Tiny, recorded comparatively greater losses, suggesting their poorer performance in anomaly detection for IoT networks. TinyML models, particularly TinyNN, were effective in resource-scarce environments. These results confirm the need to select a model on the basis of accuracy-computation trade-off, important for IoT applications where scarce resources prevail.

Model	**Accuracy**	**Precision**	**Recall**	**False Positive Rate**	**ROC AUC Score**
Random Forest	0.02	0.02	0.02	0.02	0.02
Support Vector Machine	0.18	0.18	0.18	0.17	0.18
Logistic Regression	0.04	0.04	0.04	0.04	0.04
Neural Network	0.03	0.03	0.03	0.03	0.03
Hybrid (SVM + RF)	0.03	0.03	0.03	0.03	0.03
Tiny Random Forest	0.07	0.07	0.07	0.08	0.07
Tiny Support Vector Machine	0.04	0.04	0.04	0.05	0.04
Tiny Logistic Regression	0.04	0.04	0.04	0.04	0.04
Tiny Neural Network	0.03	0.03	0.03	0.03	0.03
Tiny Hybrid Model (SVM + RF)	0.04	0.04	0.04	0.05	0.04

Table 7.1: Loss table for all metrics

Chapter 8

CONCLUSION

In summary, this project evaluated traditional machine learning models and TinyML models for anomaly detection in IoT networks based on the NSL-KDD dataset. The comparison was done using several performance parameters such as Accuracy, Precision, Recall, False Positive Rate (FPR), and ROC AUC Score. Loss values were derived by subtracting performance scores from 1, for which FPR itself was taken as loss. Across all models, however, the Tiny Neural Network (TinyNN) consistently produced the highest accuracy as well as the lowest loss measures on all indicators.

The Classical models such as Random Forest and Neural Networks also improved with little loss, which indicates their strength in the detection of anomalies. Nonetheless, the SVM models, classical and Tiny, recorded comparatively higher loss values, indicating weaker performance than the other models in this particular use case. The table of loss also corroborated this, indicating that TinyNN recorded the lowest total loss.

TinyML models, particularly TinyNN, demonstrated the ability to supply effective solutions to resource-limited environments, such as IoT networks. The models present a highly effective alternative solution to heavier typical models, matching performance and consumption. This places TinyML models as ideal options for deployment into real-time IoT systems where minimal latency and power consumption are crucial.

Future research might aim to refine these models, investigate other datasets, and enhance the scalability of TinyML models for still more advanced IoT settings. More studies on hybrid models that leverage the strengths of both TinyML and conventional models may produce more efficient solutions for anomaly detection in IoT networks.

8.1 SCOPE OF FURTHER WORK

Although the current project successfully developed an anomaly detection model for IoT networks using TinyML techniques, several areas offer potential for future improvement:

- **Hardware Deployment:** Deploy the TinyML models on Raspberry Pi and ARM-based microcontrollers for live anomaly detection testing.
- **Dataset Expansion:** Test models on real-world IoT traffic datasets beyond KDDTest+ to improve robustness and adaptability.
- **Model Enhancement:** Experiment with pruning, quantization-aware training, and knowledge distillation to improve TinyML model performance.
- **Edge Intelligence:** Develop self-learning models at the edge that adapt over time without constant retraining from cloud resources.
- **Social Impact:** Strengthen cybersecurity infrastructure across IoT deployments in critical sectors such as healthcare, agriculture, and smart cities.

These areas of future work can potentially not only enhance the performance and flexibility of TinyML models but also introduce actual- global applications that improve the security and efficacy of IoT networks from multiple industries.

REFERENCES

[1] **Friedman, J. H.**, Greedy Function Approximation: A Gradient Boosting Machine, *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[2] **Freund, Y. and Schapire, R. E.**, A Short Introduction to Boosting, *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[3] **Cover, T. and Hart, P.**, Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[4] **Mnih, V., Kavukcuoglu, K., Silver, D., et al.**, Human-Level Control Through Deep Reinforcement Learning, *Nature*, vol. 518, pp. 529–533, 2015.

[5] **Chen, T. and Guestrin, C.**, XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[6] **Hutter, F., Kotthoff, L., and Vanschoren, J.**, *Automated Machine Learning: Methods, Systems, Challenges*, Springer, 2019.

[7] **Han, J., Kamber, M., and Pei, J.**, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.

[8] **Roman, R., Najera, P., and Lopez, J.**, Securing the Internet of Things, *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[9] **Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P.**, Blockchain for IoT Security and Privacy: The Case Study of a Smart Home, *Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.

[10] **Subashini, S. and Kavitha, V.**, A Survey on Security Issues in Service Delivery Models of Cloud Computing, *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[11] **Warden, P. and Situnayake, D.**, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*, O'Reilly Media, 2019.

[12] **Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.**, Generative Adversarial Nets, *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[13] **Tankard, C.**, Big Data Security, *Network Security*, vol. 2012, no. 7, pp. 5–8, 2012.

[14] **Ahmed, M., Mahmood, A. N., and Hu, J.**, A Survey of Network Anomaly Detection Techniques, *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[15] **Esteva, A., Kuprel, B., Novoa, R. A., et al.**, Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks, *Nature*, vol. 542, pp. 115–118, 2017.

[16] **Buczak, A. L. and Guven, E.**, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection, *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[17] **Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M.**, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[18] **Aman, M. N., Chua, K. C., and Sikdar, B.**, Security of Smart Home Systems Based on Lightweight Cryptographic Authentication and Authorization, *IEEE Transactions on Consumer Electronics*, vol. 63, no. 3, pp. 330–338, 2017.

[19] **Zissis, D. and Lekkas, D.**, Addressing Cloud Computing Security Issues, *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[20] **Lee, E. A.**, Cyber Physical Systems: Design Challenges, *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008.