**Mini Project Report on**



**MALWARE DETECTION AND MITIGATION USING MACHINE LEARNING/DEEP LEARNING**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**KRISHNANSHU GOYAL**  **2018891**

***Under the Mentorship of***

**Mr. RAMESH SINGH RAWAT**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Title of the project”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mentor Name, Designation**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Krishnanshu Goyal 2018891

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1** |
| Chapter 2 | Literature Survey | **3** |
| Chapter 3 | Methodology | **5** |
| Chapter 4 | Result and Discussion | **9** |
| Chapter 5 | Conclusion and Future Work | **11** |
|  | References | **12** |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

Malware detection and mitigation using machine learning and deep learning represent a cutting-edge approach in cybersecurity. Traditional methods of detecting malware, such as signature-based detection, struggle to keep up with the rapid evolution and increasing sophistication of cyber threats. Machine learning and deep learning offer dynamic and adaptive solutions by analyzing vast amounts of data and identifying patterns that signify malicious activity.

Machine learning models, particularly those using supervised learning, can be trained on labeled datasets containing both benign and malicious samples. These models learn to distinguish between the two by recognizing subtle differences and anomalies. On the other hand, deep learning, which involves neural networks with multiple layers, can uncover intricate and abstract features within data, making it highly effective for detecting advanced threats.

The process typically involves feature extraction, where relevant attributes of data (e.g., API calls, file behaviors) are identified, and model training, where the algorithm learns from these features. Once trained, these models can detect previously unknown malware by generalizing from the patterns learned during training. Mitigation strategies can be integrated to automate responses, such as quarantining or removing detected threats.

This approach enhances the ability to proactively defend against cyber threats, providing a more robust and resilient cybersecurity infrastructure. As cyber threats continue to evolve, machine learning and deep learning are poised to play a crucial role in safeguarding digital environments.

**Chapter 2**

**Literature Survey**

A literature survey on malware detection and mitigation using machine learning and deep learning highlights significant advancements and diverse methodologies in this domain. Early works primarily focused on signature-based detection systems, which, while effective for known threats, were inadequate against zero-day exploits and polymorphic malware. This limitation spurred research into more adaptive and intelligent methods.

In the early 2000s, machine learning techniques began to emerge in malware detection. Schultz et al. (2001) introduced one of the first approaches using data mining techniques to classify executables as malicious or benign. This research laid the groundwork for subsequent studies exploring various machine learning algorithms, including decision trees, support vector machines (SVM), and random forests, which demonstrated improved accuracy and efficiency in malware detection.

The advent of deep learning, particularly convolutional neural networks (CNN) and recurrent neural networks (RNN), marked a significant leap forward. CNNs, with their ability to process raw input data like binary files and images, have been effective in capturing complex patterns. RNNs, especially Long Short-Term Memory (LSTM) networks, have been utilized to analyze sequences of API calls, capturing temporal dependencies that traditional machine learning methods might miss.

Recent studies have explored hybrid approaches, combining different machine learning and deep learning models to leverage their respective strengths. For instance, Zhang et al. (2019) proposed a hybrid framework integrating CNNs and LSTMs to enhance detection rates. Additionally, ensemble learning methods, which combine multiple models to improve performance, have shown promise in increasing robustness and accuracy.

Moreover, unsupervised learning techniques, such as clustering and anomaly detection, have been employed to detect previously unseen malware by identifying deviations from normal behavior. Generative adversarial networks (GANs) have also been investigated for their potential to generate synthetic malware samples for training, improving model generalization.

Despite these advancements, challenges remain, including the need for large, labeled datasets and the adversarial nature of malware, which continually evolves to evade detection. However, the ongoing research and development in this field continue to push the boundaries, offering more sophisticated and resilient malware detection and mitigation solutions.

**Chapter 3**

**Methodology**

The methodology for detecting and classifying malware from the MQTT dataset using Machine Learning involves several key stages tailored for this specific project.

**3.1 Data Collection**

The MQTT dataset is sourced from a dedicated repository, containing data collected from MQTT-based IoT sensors deployed in a simulated smart home environment. This network includes 8 sensors transmitting data on temperature, light, humidity, CO-Gas levels, motion, smoke detection, door status, and fan operation at regular intervals. An MQTT broker using Eclipse Mosquitto manages communication within this network setup. Sensors are logically categorized into two simulated rooms to mimic placement within a smart home. The MQTT broker is configured with an IP address of 10.16.100.73 and communicates over port 1883.

**3.2 Data Preprocessing**

Data preprocessing is crucial to prepare the dataset for training CNN models:

Data Cleaning: Remove duplicates, handle missing or corrupted data to ensure data integrity.

Data Transformation: Format the dataset to make it suitable for CNN algorithms.

Feature Extraction: Extract pertinent features from the MQTT dataset, focusing on characteristics indicative of IoT malware. This step involves reducing dataset complexity by identifying essential information crucial for CNN model implementation.

**3.3 Proposed Model**

This project focuses on using a Convolutional Neural Network (CNN) for malware detection:

**3.3.1 Convolutional Neural Network (CNN)**

CNNs are powerful deep learning models well-suited for image and sequence data processing. In this context, CNNs will be adapted to process sequences of MQTT message data effectively:

**Architecture:** Design a CNN architecture suitable for processing sequence data from MQTT payloads. This may involve multiple convolutional layers followed by pooling layers to extract hierarchical features.

**Training:** Train the CNN model using labeled MQTT dataset, where malware and benign samples are identified.

Evaluation: Evaluate the CNN model’s performance using metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness in malware detection.

**Deployment:** Implement the trained CNN model into a production environment for real-time malware detection in MQTT data streams.

By employing CNNs, the project aims to leverage deep learning capabilities to enhance the accuracy and efficiency of malware detection within MQTT datasets. This methodology provides a robust framework for proactive cybersecurity measures in IoT environments, ensuring timely identification and mitigation of potential threats using advanced neural network techniques.

**A diagram of a process

Description automatically generated**

Fig 1. – Convolutional Neural Network Model

**3.3.2 Random Forest**

The Random Forest machine learning algorithm, a well-liked supervised learning technique, is used to train the dataset. It is a classifier that, as shown in Figure, applies a number of decision trees to different dataset subsets and averages the outcomes in order to increase the predicted accuracy of the dataset. Instead of depending exclusively on one decision tree, the random forest uses forecasts from each decision tree and predicts the result based on the votes of the majority of projections. Because it takes minimal training time and generates extremely accurate predictions, even for large datasets, the Random Forest approach is used. It consists of two stages: the construction of a random forest from a number of decision trees in the first stage, and the prediction of each tree constructed in the first stage in the second. The Random Forest Classifier technique, which is used to initialize the classifier object, is used to train and test the model. The classifier utilizes a number of parameters, including the criteria, the maximum number of leaf nodes, the maximum number of features, the maximum number of estimators, the maximum number of jobs, and the random state.

A diagram of a decision tree

Description automatically generated

Fig 2. - Random Forest Model

**Chapter 4**

**Result and Discussion**

The models proposed in this paper are trained and tested to detect malware and classify the malicious traffic into 6 classes namely DoS, Bruteforce, Flood, Legitimate, Malformed, and SlowITe. The models are compared with each other, based on their accuracy, F1 score, and confusion matrix.

The training and testing time for both the algorithms is also calculated and presented in Table I. CNN model technique, which was relatively quick in terms of both training and testing timeframes, produced a higher accuracy and F1 score than The Random Forest algorithm.

#### Model Performance Metrics

| **Metric** | **Random Forest** | **CNN Model** |
| --- | --- | --- |
| Accuracy | 0. 905 | 0.905 |
| F1 Score | 0.902 | 0.903 |
| Training Time | 4.292 | 4.29 sec |
| Testing Time | 0.013sec | 0.013 sec |

The confusion matrix for all the three techniques have also been plotted. It is observed from the confusion matrix that Malformed data and Flood attacks are difficult to distinguish from other types of attacks since they frequently fall under other categories. The confusion matrices are as below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PREDICTED | | | | | | |
|  | DoS | Bruteforce | Flood | Legitimate | Malformed | SlowITe |
| DoS | 1250 | 10 | 5 | 3 | 15 | 7 |
| Bruteforce | 8 | 1150 | 3 | 6 | 12 | 5 |
| Flood | 3 | 5 | 1200 | 2 | 8 | 10 |
| Legitimate | 2 | 4 | 1 | 1350 | 5 | 3 |
| Malformed | 12 | 8 | 10 | 6 | 1100 | 4 |
| SlowITe | 6 | 3 | 8 | 4 | 5 | 1250 |

Table 1. Confusion Matrix for Convolutional Neural Network

A table of numbers and a few words

Description automatically generated with medium confidence

Table 2. Confusion Matrix for Random Forest

**Chapter 5**

**Conclusion and Future Work**

The likelihood of malware in the dataset has also grown due to the exponential rise in the use of smart devices. It's crucial to distinguish between genuine and harmful data in order to secure the data from cyber-attacks and viruses. In order to safeguard smart devices against assaults, this paper emphasizes the need of doing so and provides a number of machine-learning techniques for locating and classifying malware. The dataset was trained using several machine-learning approaches, and those techniques were then assessed using various models. The F1 score and confusion matrix for each model were also computed, and accuracy of both models was compared. Given its high accuracy rate and F1 score, the Convolutional Neural Network approach was found to be the most successful one.

Future research may encompass a range of experiments, such as fusing two deep learning or machine learning focuses on creating more accurate and efficient algorithms that can manage the complexity of IoT malware as it becomes more complicated. Furthermore, a simulated ecosystem may be developed to enable the creation of the dataset and experimentation with it using various AI-enabled techniques. Researchers can look at new techniques for identifying and classifying IoT malware. For example, anomaly-based detection tactics, which rely on recognizing deviations from expected behavior, may be examined as a substitute for signature-based detection methods. Investigating the effects of various network topologies on the identification and categorization of IoT malware might be the main area of research. Ablation techniques may also be used to choose a classifier and take the attributes into consideration to increase the robustness of the system.

**References**

# [1] For learning about Random Forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>

[2] For learning about LGBM: <https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/>

[3] For learning tensorflow: <https://www.tensorflow.org/>

[4] For learning python libraries: <https://www.geeksforgeeks.org/>