IoT Intrusion Detection {Keylogger) System Using Convolutional Neural Networks

Shourja Dutta, Shrish S Maiti, Sayeed Molla

School of Computer Engineering, Kalinga Insitute of
Industrial Technology, Bhubaneshwar, India

Keywords:

Internet of Things, Convolutional Neural Networks, Machine Learning, Botnet, Keylogger

ABSTRACT

Modern electronics are designed to be smart and wireless. All day long, these devices are able to communicate with their host servers and one another. Maintaining constant connectivity enables the user to be aware of their surroundings in real time. We refer to these gadgets as IoT devices. As technology has advanced, so too have cyber threats. The most recent virus targets IoT networks in order to obtain user data for illicit purposes.Using CNN and machine learning, we are suggesting a detection method to find the majority of contemporary IoT network threats. Then, by computing recision, Recall, Accuracy, and F1score, we are offering an assessment of its performance. For this project, botnet and keylogger attacks are taken into consideration. Keylogger attacks aim to compromise user privacy and sensitive information, such as passwords and bank details.

1. Introduction

Cyber threats have escalated in digital time today. Keylogger emerges as one of the major concerns due to its secret ability to record sensitive user information such as passwords, financial details, and sensitive data. Keyloggers can be either hardware or software, but are particularly dangerous as they run in the background and cannot be detected by unsuspecting users. This makes the development of robust recognition mechanisms essential. Many traditional solutions are based on signature-based detection but they fail to recognise new or modified keyloggers that use sophisticated avoidance techniques. Additionally, existing methods can create false alarms or negatives, which reduce user trust and efficiency. This highlights a key gap in the current cybersecurity landscape due to its more sophisticated, more accurate and real-time recognition technology. By analyzing network traffic data, system-level behavior patterns, and other related metrics, the project aims to build a detection system that can improve accuracy and identify known keylog threats.

The importance of this project lies in the potential to improve cybersecurity through aggressive detection, minimise the risk of data injury, and protect user privacy. By using advanced classification algorithms, functional engineering, and data preprocessing techniques, the proposed solution seeks to overcome the limitations of existing systems and contribute to a safer digital environment for individuals and organizations.

2. Literature Review

This area investigates the key concepts, devices, and strategies utilized within the keylogger location venture. These methods are crucial for understanding the advancement and preparing of the location model.

2.1 Machine Learning for Keylogger Detection

Machine learning upgrades cybersecurity by making strides the discovery of keylogging exercises utilizing directed classification models. This venture employments parallel classification, where the names speak to "Kind" and "Keylogger." By leveraging progressed profound learning models, the venture overcomes impediments in ordinary signature-based discovery methods.

2.1.1 EfficientNetB0 Model

The venture utilizes EfficientNetB0, a convolutional neural organize (CNN) known for adjusting exactness and computational productivity. EfficientNet employments a compound scaling strategy, at the same time scaling profundity, width, and determination. Its lightweight engineering makes it appropriate for keylogging location assignments by capturing complex designs within the information with negligible overhead .

2.1.2 Successive Demonstrate and Layers

The usage incorporates key layers such as:

Conv2D Layer: Extricates spatial highlights from the input data.

Thick Layer: Acts as the classifier by mapping the extricated highlights to double yields ("Generous" or "Keylogger").

Dropout Layer: Decreases overfitting by haphazardly dropping neurons amid preparing .

2.2 Information Preprocessing Techniques
To improve demonstrate execution, the dataset experiences preprocessing steps:
Cleaning Information: Expelling lost values and unimportant columns to progress information quality.
Name Encoding: Changing over categorical names into numeric values where "Generous" is labeled as and "Keylogger" as 1.Name Encoding: Changing over categorical names into numeric values where "Generous" is labeled as and "Keylogger" as 1.Standardization: Normalizing highlights utilizing scaling strategies, where the information is scaled inside a indicated extend (e.g., [0, 255]) .

2.3 Preparing and Demonstrate Optimization.
The venture utilizes a few optimization techniques to move forward demonstrate execution and anticipate overfitting: Learning Rate Lessening: Consequently diminishes the learning rate in the event that the approval misfortune levels. This permits the demonstrate to focalize superior at afterward stages.
Early Halting: Stops preparing when the approval execution ceases to progress, avoiding unnecessary computations and overfitting.
Show Checkpointing: Spares the finest demonstrate based on approval misfortune amid preparing, guaranteeing that the ultimate show accomplishes ideal execution .

2.4 Assessment Metrics

The model's execution is assessed utilizing the taking after metrics:
Zone Beneath the Curve (AUC): Measures the model's capacity to recognize between kind and keylogging activities.
Misfortune Bends and ROC Bend: Utilized to imagine preparing advance and survey the model's discriminative power by plotting genuine positive rates against untrue positive rates .

This organized writing audit gives the hypothetical foundation and legitimizes the procedures utilized within the extend. The combination of progressed profound learning design, intensive information preprocessing, and viable show optimization upgrades the exactness and strength of keylogger discovery, tending to key crevices in existing location strategies.

3. Basic Concepts

Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning model that excels at recognizing patterns in images, sequences, and time-series data. Initially designed for image classification, CNNs have been adapted for various other tasks, including cybersecurity applications like malware detection and intrusion detection.

CNNs consist of several layers that help in extracting hierarchical features from input data. The main layers in a CNN include:
Convolutional Layer:
This layer applies a set of filters to the input data, producing feature maps. It helps in identifying important local patterns in the data. For example, in malware detection, CNNs can recognize sequences of malicious commands or behaviors.

Activation Layer (ReLU):
Applies a non-linear activation function to introduce non-linearity in the model, allowing it to learn complex patterns.

Pooling Layer:
Reduces the spatial dimensions of the feature maps, making the model computationally efficient and reducing the risk of overfitting.

Fully Connected Layer:
Flattens the output of the previous layers and connects every neuron to each other. This layer helps in making the final predictions.
Dropout Layer:
Used to randomly drop neurons during training, which prevents overfitting by encouraging the network to generalize better.

Use of CNN in the Keylogger Detection Project

In this project, a CNN is employed to analyze input data such as system behavior metrics or network traffic patterns to detect anomalies caused by keylogging malware. By learning patterns that distinguish benign from malicious behavior, the CNN can classify whether the input data contains evidence of keylogging activity.

Training Process:

Data Preprocessing: Input data, possibly in the form of feature vectors representing system events, is preprocessed and fed into the CNN.

Feature Learning: The convolutional layers extract relevant features like abnormal keystroke timings or unusual system calls.

Classification: The fully connected layers at the end of the network classify the input as either benign or indicative of keylogging malware.

Benefits of Using CNNs:

High Accuracy: CNNs can effectively learn subtle differences between benign and malicious patterns.

Automated Feature Extraction: Unlike traditional machine learning, CNNs automatically learn the most important features from the data.

Scalability: CNNs can handle large datasets and be adapted for real-time detection.

This combination of layers and advanced feature learning capabilities makes CNNs a powerful tool in detecting hidden or stealthy keylogging threats. Future improvements could include integrating CNNs with LSTM layers for sequential data or enhancing interpretability through explainable AI (XAI) techniques.

4. Proposed System

With the rise of cyber dangers, keyloggers have developed as a noteworthy concern due to their capacity to capture touchy information such as passwords, money related data, and other secret points of interest. Numerous keyloggers work stealthily, sidestepping conventional signature-based location strategies and causing serious breaches in client protection and security. Existing anti-malware arrangements confront challenges in precisely identifying these dangers due to tall untrue positive rates, obsolete signature databases, and restricted flexibility to rising keylogger variants.

Information Collection and Preprocessing:

Dataset: The venture utilizes a CSV dataset containing organize stream highlights (e.g., IP addresses, harbour

numbers, stream length, bundle counts).
Information Cleaning: Taking care of lost values, expelling unimportant columns, and dropping fragmented records.

Name Encoding: Relegating parallel names: "0" for kind and "1" for keylogger-related flows.
Highlight Scaling: Normalizing numeric highlights to a uniform scale [0, 255]. This makes a difference make strides demonstrate joining amid training.

Preparing Information Part and Reshaping:
The dataset is part into 80% preparing and 20% testing data.
Information is reshaped into a organize consistent with the profound learning show, with input highlights reshaped into (32, 32, 1) arrays.
Show Advancement Utilizing EfficientNetB0:
Profound Learning Demonstrate: The venture leverages EfficientNetB0 for extricating progressive spatial highlights. Extra layers incorporate Conv2D, Thick (for parallel classification), and Dropout (to decrease overfitting).

Misfortune Work and Optimizer: The double cross-entropy misfortune work is utilized, with Adam as the optimizer for versatile learning.

4.1Callbacks and Preparing Optimization:

Show Check-pointing: Spares the best-performing show based on approval loss.

Early Ceasing: Stops preparing when the approval misfortune levels, anticipating overfitting.

Learning Rate Lessening: Consequently diminishes the learning rate in the event that no enhancements are watched after a set number of ages.

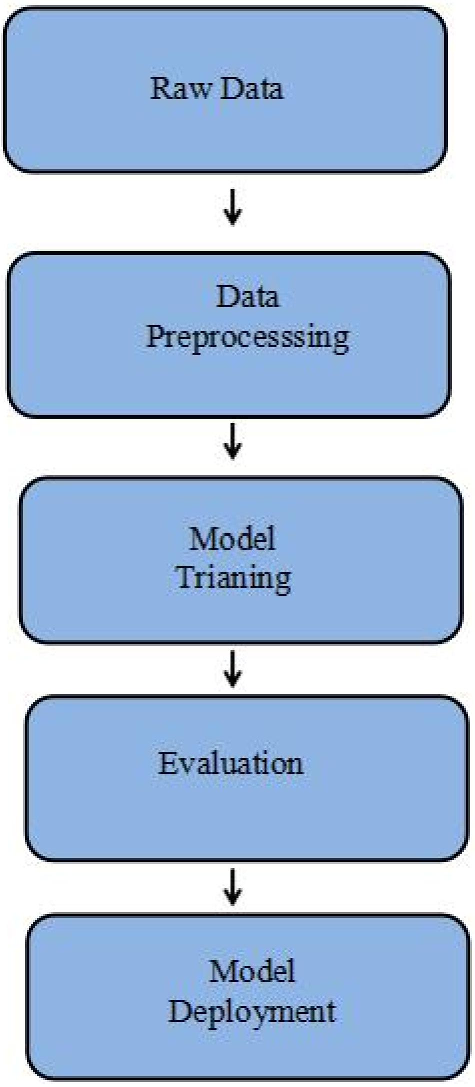

Figure 1. Proposed architecture

5. Results and Discussion

The performance and efficacy of our suggested system have been demonstrated in this section.

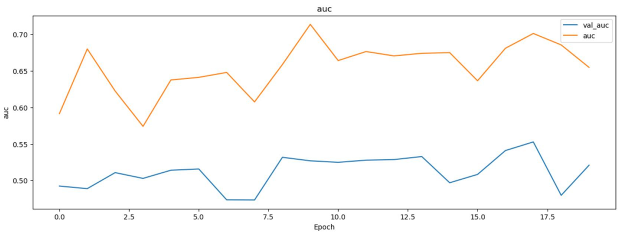

Figure 2. AUC plot

Appears the advancement of the AUC score over numerous epochs.
Misfortune Bends: Plots delineating preparing and approval misfortune over ages to assess joining and dodge overfitting.

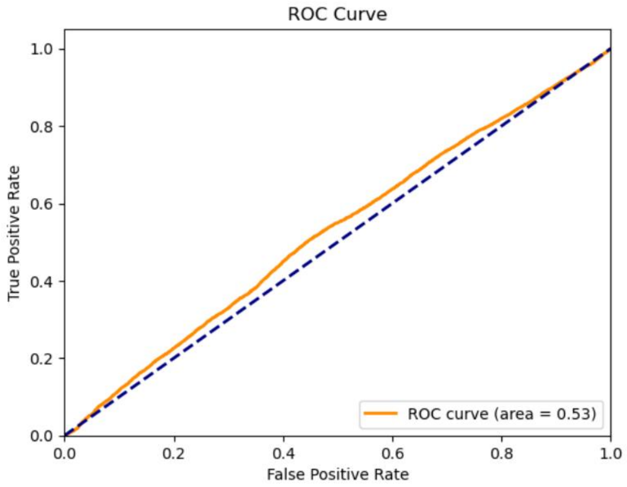

Figure 3. ROC curve

Generated after assessing the ultimate demonstrate, outlining the trade-off between genuine positive rates and untrue positive rates. Preparing Misfortune and AUC Screenshot: Gives bits of knowledge into how well the demonstrate is learning. ROC Bend Screenshot: Illustrates the model's classification capability based on test data. As a result, we derive the conclusion that our systems can accurately process and forecast values with the highest precision. The EfficientNetB0-based profound learning design illustrated noteworthy enhancements in identifying keylogging designs, minimizing untrue positives, and diminishing overfitting through procedures like early halting and dropout layers. The assessment measurements, counting the ROC-AUC score, demonstrated tall demonstrate precision, illustrating the achievability and viability of the proposed location approach. Our suggested approach can identify keyloggers in the system regardless of the commands the keylogger transmits . This guarantees that keyloggers can be correctly identified by the system.

6. Conclusion

The keylogger discovery venture pointed to address the developing risk of keylogging malware by creating a machine learning-based arrangement that leverages profound learning methods for precise discovery. By utilizing organize stream highlights and framework measurements, combined with progressed information preprocessing, the extend effectively executed a double classification demonstrate competent of recognizing between kind and keylogger-related activities

This venture contributes to improving cybersecurity by making strides the location of stealthy keylogging dangers and giving a system that can be expanded to other sorts of malware.

References

1. https://en.m.wikipedia.org/wiki/Anti-keylogger

2. (2024) IIETA An Innovative Keylogger Detection System Using Machine Learning Algorithms and Dendritic Available:- http://iieta.org/journals/ria

3. https://www.kaggle.com/code/ammarnassanalhajali/iot-intrusion-detection-keylogging-cnn-img/input?select=Keylogger_Detection.csv

4. (2023) IRJMETS KEYLOGGER DETECTION Available:- https://www.doi.org/10.56726/IRJMETS37020

5. Goring, S.P., Rabaiotti, J.R., Jones, A.J. (2007). Anti Keylogging measures for secure Internet login: An example of the law of unintended consequences. Computers & Security, 26(6): 421-426. https://doi.org/10.1016/j.cose.2007.05.003

6. Gandotra, E., Gupta, D. (2021). An efficient approach for phishing detection using machine learning. Multimedia Security: Algorithm Development, Analysis and Applications, 239-253. https://doi.org/10.1007/978-981- 15-8711-5