**Hybrid Deep Learning for Botnet Attack Detection in the Internet of Things Networks**

**Abstract:**

Detecting botnet attacks in memory-constrained Internet of Things (IoT) devices poses a significant challenge due to the large volume of network traffic data and the associated memory requirements of Deep Learning (DL) methods. This paper presents an innovative approach to address this challenge by leveraging the encoding phase of a Long Short-Term Memory Autoencoder (LAE) to reduce the feature dimensionality of large-scale IoT network traffic data. The low-dimensional feature set generated by LAE is then analyzed using a deep BoT-Convolutional Neural Network with Long Short-Term Memory (BT-CNNLSTM) to classify network traffic samples accurately. Experimental evaluations are conducted using the BoT-IoT dataset to validate the effectiveness of the proposed hybrid DL method. The results BT-CNNLSTM model exhibits robustness against model under-fitting and overfitting, showcasing its effectiveness in binary and multi-class classification scenarios.

**Hardware Requirements**

* System: Intel core I3 3.80 GHz 64 bit.
* Monitor: LED.
* Mouse: Logitech.
* Ram: 4.00 GB.

**Software Requirements:**

* Operating system : Windows 10
* Platform : Anaconda3
* Development Environment : Jupyter notebook
* Programming Language : python

Modules:

* Dataset Collection:
* Pre-processing
* Feature extraction
* BT-CNNLSTM
* Performance Evaluation

Module Description:

Dataset Collection

The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of UNSW Canberra. The network environment incorporated a combination of normal and botnet traffic.  The csv files were separated, based on attack category and subcategory, to better assist in labeling process. This dataset download from kaggle.

Pre-processing:

The Large network traffic data in Bot-IoT dataset was preprocessed to transform the features and the ground truth labels into appropriate formats for ease of computation. In this study, data pre-processing involves the following: (a) elimination of redundant network information; (a) random division of complete network traffic data into training, validation and testing sets; (b) selection of network traffic features and ground truth labels; (c) normalization or scaling of network traffic features; and (d) integer encoding of ground truth labels

Feature extraction:

A Linear Autoencoder (LAE) is a neural network architecture designed for feature dimensionality reduction. In the context of botnet dataset analysis, LAE serves to encode and compress high-dimensional feature representations into a lower-dimensional space. The model consists of an encoder network responsible for mapping input features to a reduced representation and a decoder network that reconstructs the original features from this compressed representation. By training the LAE on botnet data, the network learns to capture essential patterns and features while discarding redundant information. The choice of the bottleneck layer's dimensionality is critical, as it determines the level of compression. Post-training, the LAE allows for transforming the input data into a more compact representation, aiding in visualization, anomaly detection, or downstream tasks such as botnet detection. Experimentation and careful evaluation are crucial for optimizing hyperparameters and ensuring that the reduced dimensions effectively encapsulate relevant information for meaningful analysis and interpretation.

BT-CNNLSTM:

A Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) for classification merges the spatial feature extraction capabilities of CNNs with the sequential modeling proficiency of LSTMs. Convolutional layers in the CNN part are adept at capturing local patterns, while LSTM layers excel in learning long-term dependencies in sequential data. In the context of classification tasks, such as time series or sequential data associated with botnet detection, the CNN-LSTM architecture learns hierarchical representations from the input data. The initial convolutional layers extract spatial features, and the LSTM layers process the sequential information. The combined architecture allows the model to effectively capture both local and long-range dependencies, contributing to improved performance in classifying complex temporal patterns, making it particularly suitable for tasks like botnet detection where understanding both spatial and temporal aspects is crucial. Fine-tuning and optimizing hyperparameters are essential for achieving optimal performance in such a hybrid model.

**Performance Evaluation**

For binary classification tasks, there are performance metrics:

1. **Accuracy:**
   * The ratio of correctly predicted instances to the total instances. It provides a general overview of the model's correctness.



1. **Precision (Positive Predictive Value):**
   * The ratio of correctly predicted positive observations to the total predicted positives. It measures the accuracy of the positive predictions.



1. **Recall (Sensitivity, True Positive Rate):**
   * The ratio of correctly predicted positive observations to all actual positives. It measures the model's ability to capture all positive instances.

​

1. **F1 Score:**
   * The harmonic mean of precision and recall. It provides a balanced measure between precision and recall.



**SYSTEM ARCHITECTURE DIAGRAM:**

Training Set

BoT-IoT Dataset Collection

Preprocessing

LAE (Feature Reduction)

Dataset Splitting

Testing Set

BT-CNNLSTM Model Build

Training

Testing

Performance Evaluation

Flow Diagram:

Dataset Collection

Preprocessing

LAE Feature Reduction

Dataset Splitting

BT-CNNLSTM Training

Testing and performance Evaluation

Data Flow Diagram:

