SUMMER INTERN PROJECT PRESENTATION

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR
ASSAM

Malware Hunter: A CNN powered malicious URL detection system

Presented By:

Anand Kumar 2013008

Mula Ganesh 2013067

Under the Guidance of:

Dr. Ripon Patgiri

Assistant Professor

Department of Computer Science and Engineering

National Institute of Technology Silchar

INTRODUCTION

- Malicious URLs are one of the biggest threats to this digital world and preventing it is one of the challenging tasks in the domain of cyber security.
- Previous research to tackle malicious URLs using hard-coded features have proven good indeed, but it comes with the limitation that these features are non-exhaustive and therefore detection algorithms fail to recognize new or unseen malicious URLs.
- However, with the deep learning revolution, this problem can be easily solved, since deep learning models extract features of their own by learning from patterns occurring in such URLs.

LITERATURE SURVEY:

Research Article	authors	Research Findings
1. Deep Approaches on Malicious URL Classification	Arijit Das , Ankita Das , Anisha Datta, Shukrity Si and Subhas Barman	• The CNN LSTM hybrid model is trained for 120 epochs using preprocessed URLs and their corresponding class labels. • The model is validated on a test set of 58,440 URLs. • The CNN LSTM model achieves an accuracy of 93.59%.
2. Malicious URL Detection using Deep Learning	R, vinayakumar; S, Sriram; KP, Soman; Alazab, Mamoun	• Objective is to classify whether the URL is either benign or malicious. • Character-level embedding methods were used for text representation. • Most of the models performed well on Data set 1 in comparison to Data set 2 random split and Data set 2 time split.

OBJECTIVES:

- To develop a CNN model for malicious URLs classification that is accurate and efficient.
- To evaluate the performance of the proposed model on a real-world dataset.
- To identify the limitations of the proposed model and suggest directions for future work.

METHODOLOGY:

- Collect a dataset of malicious and benign URLs.
- Preprocess the data by tokenizing the URLs by characters and padding them to a fixed length.
- Train a CNN model on the training data using the Adam optimizer and the sparse categorical crossentropy loss function.
- Evaluate the model on the testing data.

Dataset:

- The dataset is loaded from a CSV file containing two columns: 'url' and 'label'.
- 'url' column contains the URLs to be classified.
- 'label' column contains the corresponding labels ('phishing' , 'benign', 'defacement' and 'malware').
- The URLs were collected from a variety of sources, including public blacklists, phishing websites, and legitimate websites.

URL	***Label***
gurl.com/category/your-life	Benign
lazada.co.id/sanken-official-store	Benign
codeweavers.com/account/downloads	Benign
vvorootad.top/admin.php?f=1.dat	Malicious
fryzjer.elblag.pl/dfr/sercurity.htm	Malicious
keepgrowing.net.br/sial/New%20folder%20file	Malicious

Fig1: Sample dataset

DATA PREPROCESSING:

- **Label Conversion:** We converted the labels to numerical format so that the model could better understand them.
  - Benign as 0
  - Defacement as 1
  - Phishing as 2
  - Malware as 3
- **Data Splitting:** We then split the dataset into training and testing sets to prevent overfitting and to assess the model's performance effectively.
  - Train data contains 80% of dataset
  - Test data contains 20% of dataset

CONTINUATION:

- **Tokenize The URLs:** The URLs are tokenized by characters using the Tokenizer from Keras.

"https://leetcode.com/problems/single-element-in-a-sorted-array/description/"

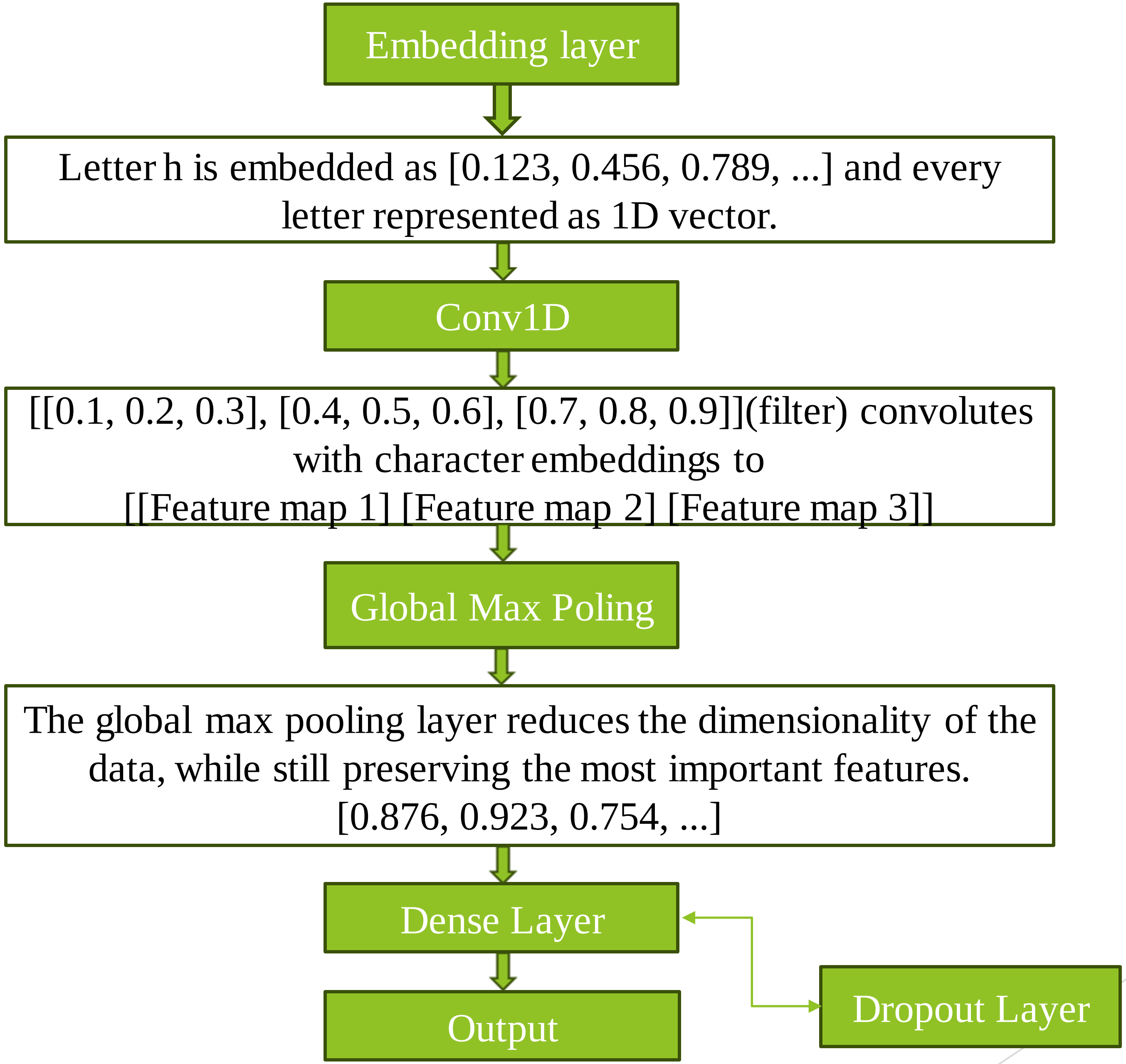
Tokenized URL: ['h', 't', 't', 'p', 's', ':', '/', '/', 'l', 'e', 'e', 't', 'c', 'o', 'd', 'e', '.', 'c', 'o', 'm', '/', 'p', 'r', 'o', 'b', 'l', 'e', 'm', 's', '/', 's', 'i', 'n', 'g', 'l', 'e', '-', ……]

- **Sequence Padding:** The sequences are padded to a fixed length (maxlen=100) to ensure uniform input shape for the CNN model.

MODEL ARCHITECTURE:

The model consists of an Embedding layer, followed by a 1D Convolutional layer, Global Max Pooling, and Dense layers.

- **The Embedding layer:** It learns the representation of each character in the URL.
- **The Conv1D layer:** It performs convolutions over the character embeddings to capture local patterns.
- **Global Max Pooling layer:** It extracts the most important features from the convolutional layer.
- **Dense layers:** This layer with ReLU activation and Dropout are used for classification.
- **Dropout Layer:** This layer randomly drops 50% of the neurons during training. This helps prevent overfitting by reducing the reliance on specific neurons.


Embedding layer
Letter h is embedded as [0.123, 0.456, 0.789, ...] and every letter represented as 1D vector.
Conv1D
[[0.1, 0.2, 0.3], [0.4, 0.5, 0.6], [0.7, 0.8, 0.9]](filter) convolutes with character embeddings to [[Feature map 1] [Feature map 2] [Feature map 3]]
Global Max Poling
The global max pooling layer reduces the dimensionality of the data, while still preserving the most important features. [0.876, 0.923, 0.754, ...]
Dense Layer
Dropout Layer
Output

Model Training:

- The model was trained on the training data using the Adam optimizer and the sparse categorical cross-entropy loss function.
- The model was trained for 10 epochs.

Model Testing:

- The model was evaluated on the testing data.
- The model achieved an accuracy of 98% on the testing data.

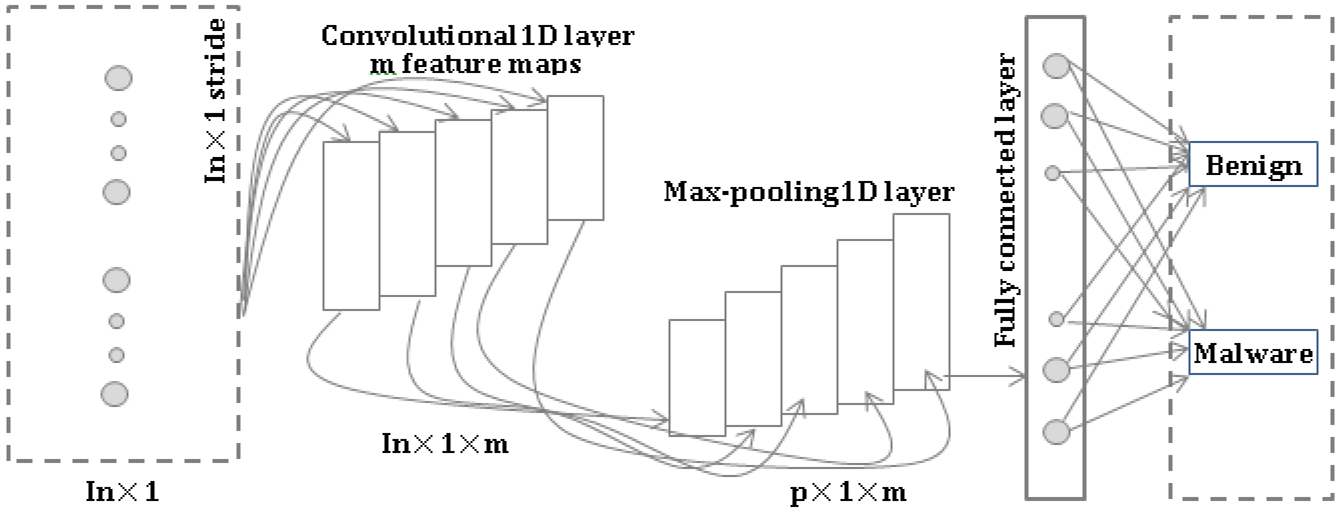


Fig2: Model Architecture

RESULTS:

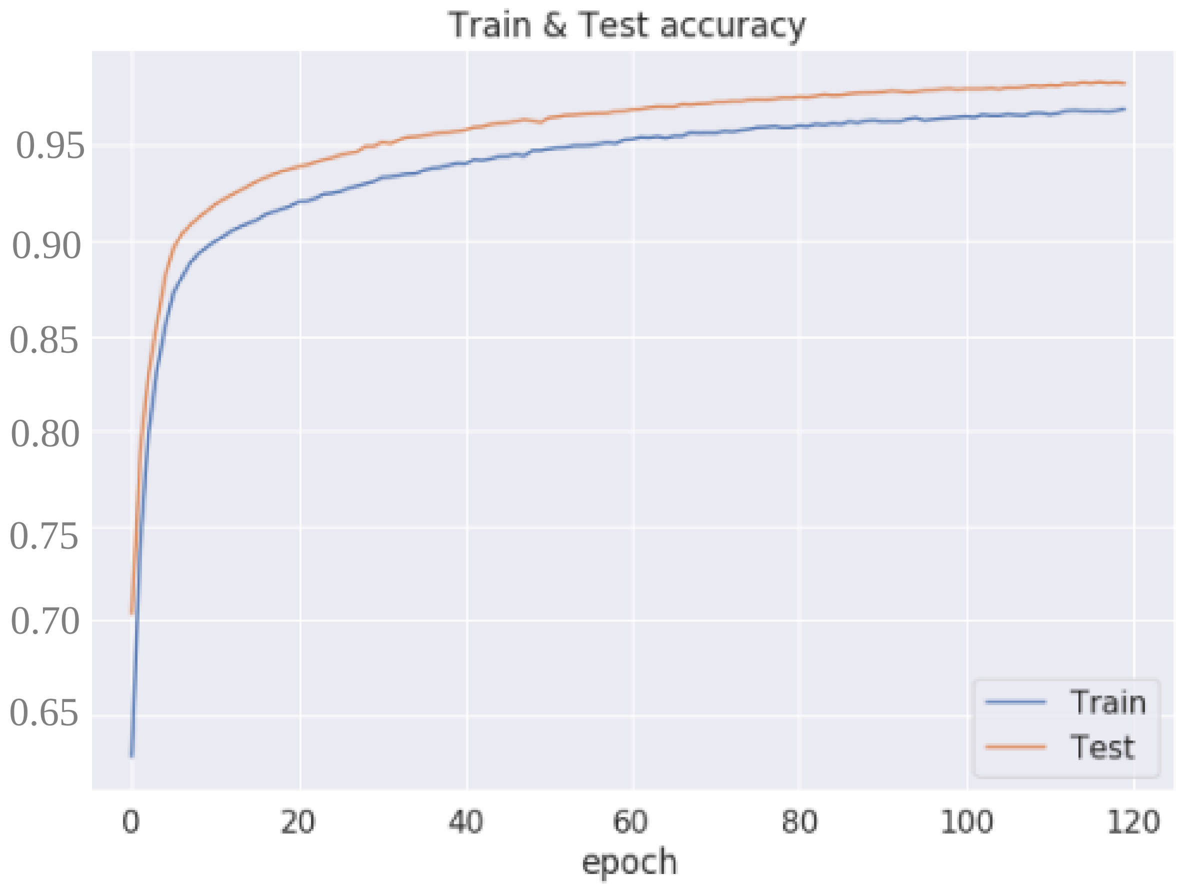


Fig3: Train & Test accuracy graph

CONCLUSION:

- The CNN model is effective for malicious URL detection. The model achieved a high accuracy on the testing dataset, demonstrating its ability to generalize to new data. The model is also relatively simple and can be easily implemented, making it a suitable model for use in real-world applications.

THANK
YOU