Malicious URL Classification - A Comparative Study

Prasitaa Kannadasan 2022115014
Trisha Rajesh 2022115132
Ananyaa Sivakumar 2022115135

Introduction

- With the rise in internet usage, malicious URLs have become a significant threat to cybersecurity. These URLs often lead to phishing sites, malware downloads, or defacement attempts.

- Existing signature-based detection systems can only detect known threats, leaving users vulnerable to new or evolving attacks.

- Our project proposes a deep learning-based model that can automatically classify URLs as benign, phishing, malware, or defacement based on various features extracted from URLs.

Problem Definition

- **Objective**: Develop a machine learning model and deep learning model to classify URLs as benign, defacement, malware, or phishing, helping to enhance cybersecurity by identifying potentially harmful URLs.

- **Approach**: Extract features from URLs, apply preprocessing techniques, and test multiple classification algorithms (Logistic Regression, Decision Tree, Random Forest, KNN, and SVM) to find the most accurate model for multiclass URL classification.

- **Outcome**: Compare the performance of each model based on accuracy, training time, and F1 scores to determine the best approach for reliable URL classification.

Objective

- The primary objective is to develop a robust and accurate deep learning-based system that can identify and classify malicious URLs.
- The model should be able to differentiate between benign, phishing, malware, and defacement URLs based on extracted features.
- The system should be capable of processing real-time inputs and generalizing well across unseen, novel threats.
- Ensure that the solution is scalable and can handle large datasets for comprehensive detection.

Ideas

Models:

- SVM and k-NN: Simple classifiers chosen to evaluate initial performance. SVM is effective for **structured data** and often **performs well in high-dimensional spaces like URL feature**s.
- Random Forest: Combines multiple decision trees, improving stability and accuracy, particularly **useful for complex datasets**.
- Deep Learning: Suitable for complex pattern recognition due to its **ability to learn hierarchical features directly from raw data.**

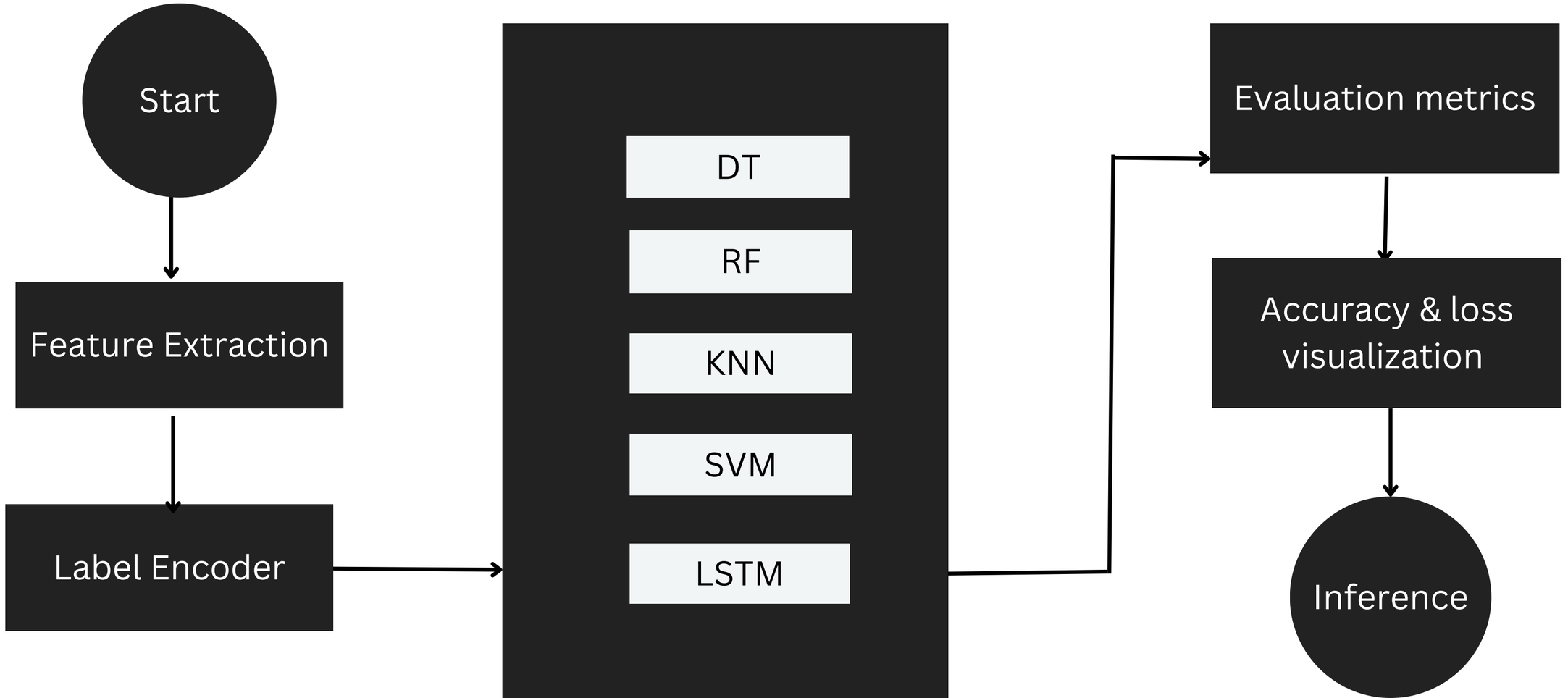
Instance Selection:

- DRLSH (Dimensionality Reduction with Locality-Sensitive Hashing) : Select representative samples based on similarity among data points to improve performance.
- Random selection: To optimize resource usage and reduce data redundancy

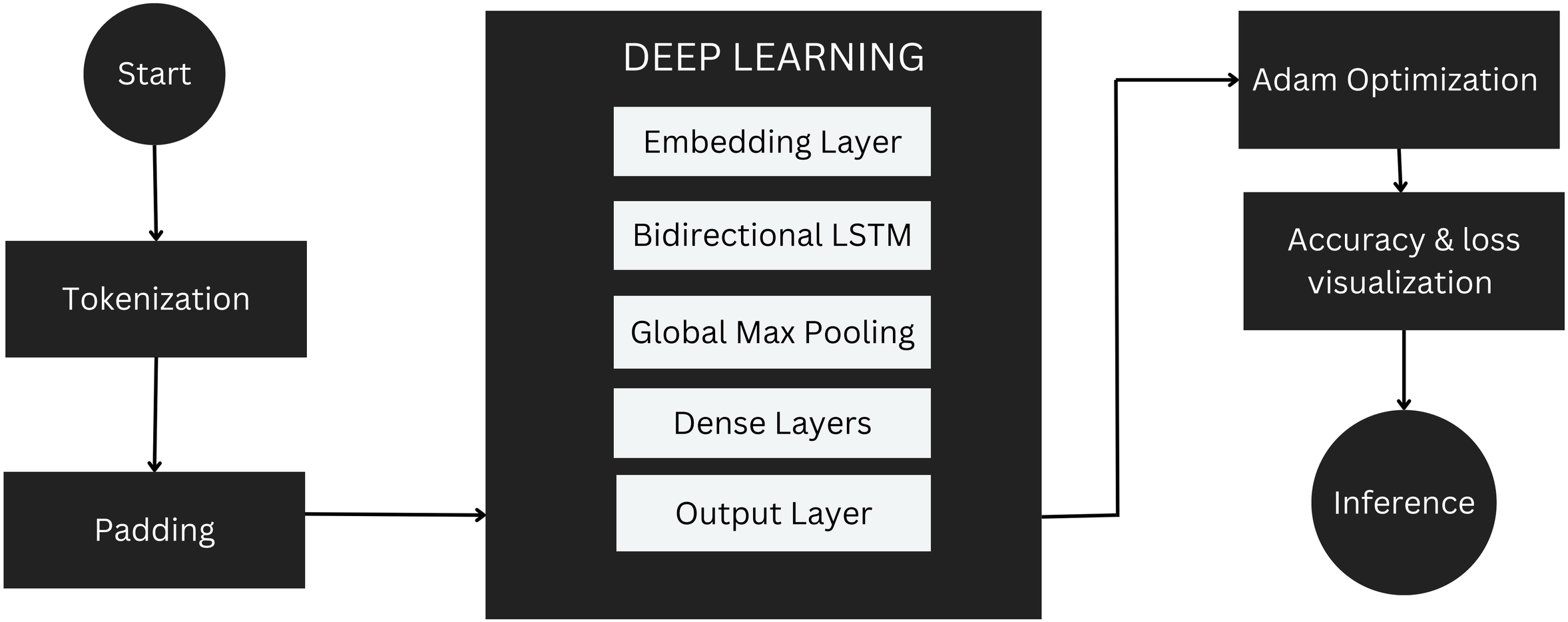
Ideas

- Deep learning, can discover new patterns autonomously, with its ability to learn hierarchical representations from raw data, presents an optimal solution for this problem.
- Multi-Layer Perceptron (MLP) : Feedforward neural network was chosen for its simplicity and efficiency in handling structured data derived from URL features.
- A **Bidirectional** LSTM layer, an advanced **RNN layer** that captures sequential URL patterns, which is then pooled for dense classification layers.
- The system is deployed using Flask, with the model running in an Anaconda environment.

System Architecture



Detailed Design (for DL)



Detailed Design

Data Preprocessing : Clean and normalize data, extract features via tokenization and vectorization, and address class imbalance with SMOTE.

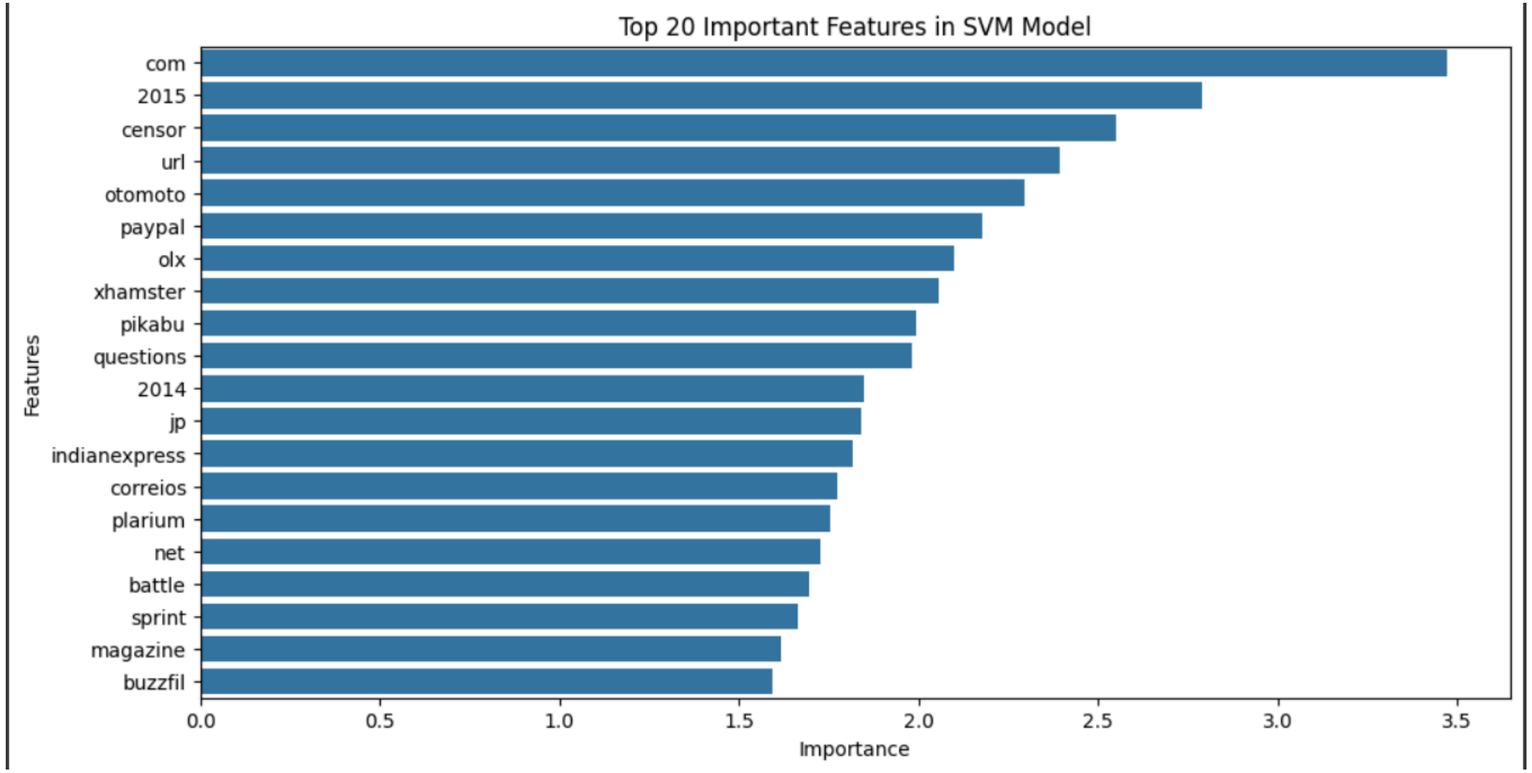
Model Architecture :
- Traditional ML: SVM, k-NN, Random Forest.
- Deep Learning: Fully connected neural network or LSTM.

Training : Split dataset, optimize with early stopping, and evaluate using precision, recall, F1-score, and confusion matrix.

UI : Developed a responsive and user-friendly web interface using HTML and CSS integrated with Flask, enabling users to input URLs for classification and to view predictions.

Future Work : Natural Language Processing features like sentiment analysis or keyword extraction could enhance the model's understanding of URL context, further improving accuracy.

Detailed Design



The features used for classifying the URL is depicted by the above graph.

Tools and Technologies Used

- Programming Language: Python

- Libraries:
  - TensorFlow/Keras: Deep learning frameworks used to build and train neural networks.
  - Scikit-learn: For SVM, k-NN, RF, and evaluation metrics.
  - Pandas/NumPy: For data preprocessing and statistical analysis.
  - DRLSH and Random Selection: Implemented as custom instance selection methods using locality-sensitive hashing (LSH).

- Dataset: Over 600,000 URLs, comprising benign, phishing, malware, and defacement classes.

Results and Discussions

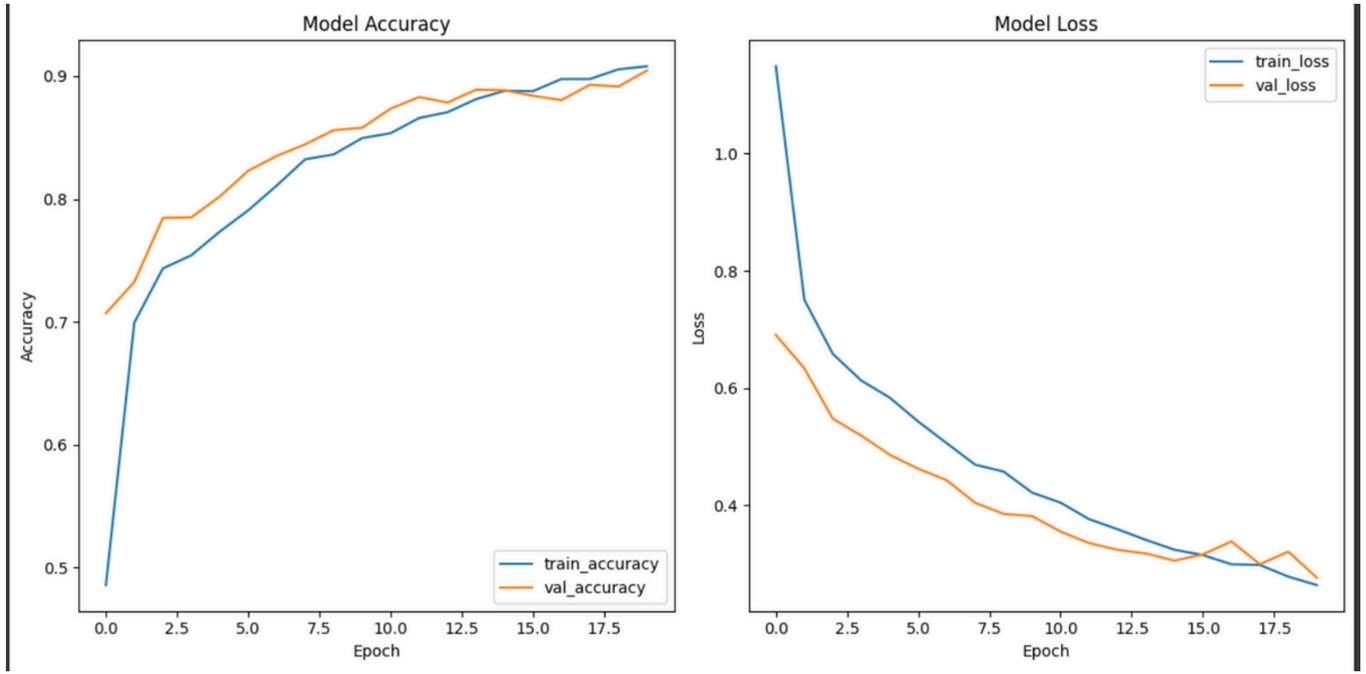
- SVM with random selection achieved the highest precision and F1 scores (94%) across all models even though it was computationally intensive
- SVM with DRLSH method showed 74% accuracy, reducing computation by focusing on essential data samples.
- DT and RF achieved an equal accuracy of 85%.
- KNN showed lower performance when using the BPLSH feature selection with about 83% accuracy.

Results and Discussions

Deep Learning:

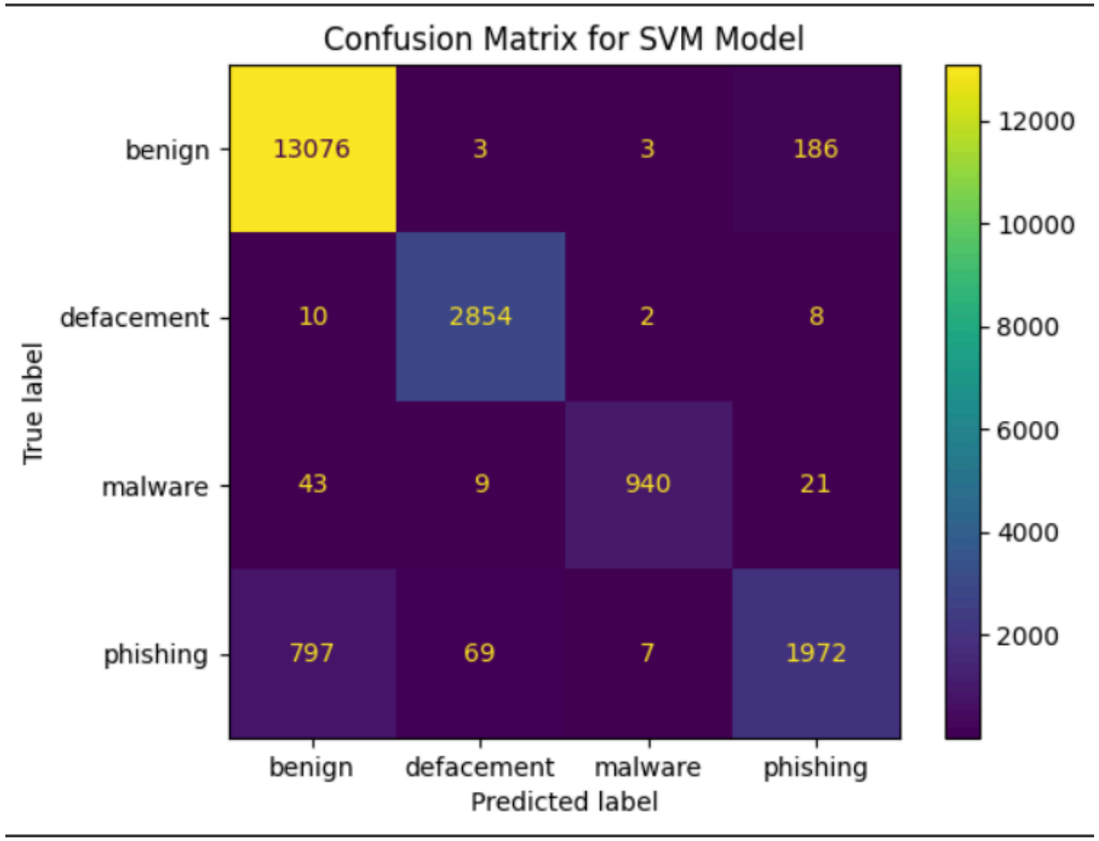
- Best accuracy (90.45%) but computationally intensive.
- Character-level tokenization allows the model to learn nuanced patterns in URLs.
- Bidirectional LSTM and RNN layers enhance feature extraction.
- The use of advanced instance selection techniques reduced computational costs without affecting model performance.

Results and Discussions



- Consistent Loss Reduction as both training and validation losses decrease steadily
- The close alignment between training and validation losses, indicate minimal overfitting

Results and Discussions



Results and Discussions

Model	Accuracy	Training Time (seconds)	F1 Score (Weighted Avg)
Logistic Regression	0.70	12.73	0.65
Decision Tree	0.85	1.96	0.84
Random Forest	0.85	52.55	0.85
K-Nearest Neighbor	0.83	30.48	0.83
SVM (Random Selection)	0.94	354.0	0.94
SVM (DRLSH-selected features)	0.74	308.4	0.75

A comparative study of all models using metrics like Accuracy, F1 score and Training Time.

References

- Gupta, B.B.; Yadav, K.; Razzak, I.; Psannis, K.; Castiglione, A.; Chang, X. A Novel Approach for Phishing URLs Detection Using Lexical Based Machine Learning in a Real-Time Environment. Comput. Commun. 2021, 175, 47–57.

- Veale, M.; Brown, I. Cybersecurity. Internet Policy Rev. 2020, 9, 2

- Sara Afzal,Muhammad Asim, Abdul Rehman Javed, Mirza Omer Beg & Thar Baker : URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models