

AI-Driven Smishing Detection in Android Devices Using TinyBERT and Aquila Optimization

Akshat Gaurav¹, Brij B. Gupta^{2*}, Kwok Tai Chui³

¹Ronin Institute, Montclair, NJ, USA

²Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan

³Hong Kong Metropolitan University, Hong Kong SAR, China

akshat.gaurav@ieee.org, bbgupta@asia.edu.tw, jktchui@hkmu.edu.hk

Abstract—The increasing prevalence of smishing (SMS phishing) attacks on mobile devices poses significant security risks, particularly for Android users. Traditional spam detection techniques often fail to accurately distinguish between legitimate and malicious messages due to the complexity and contextual nature of smishing. In this paper, we propose an AI-driven smishing detection approach using TinyBERT for feature extraction and Aquila Optimization (AO) to optimize a deep learning model. With AO-tuned hyperparameters, the proposed model achieved an accuracy of 96.81%, outperforming standard models like GRU, LSTM, and SVM. Our approach offers a robust, efficient solution for Android smishing detection.

Keywords—Smishing Detection, TinyBERT, Aquila Optimization, Android Security, SMS Phishing

I. INTRODUCTION

The proliferation of smartphones has significantly transformed communication and information access, leading to an increased vulnerability to various cybersecurity threats, particularly smishing (SMS phishing) attacks. As smartphones have become ubiquitous, with billions of users globally, they have also become prime targets for cybercriminals. The rise in mobile device usage has expanded the attack surface for malicious activities, including phishing, malware, and unauthorized access to sensitive information[1, 2, 3]

Smishing attacks exploit the convenience of SMS, a communication method that is widely used for personal, financial, and business purposes. Research indicates that these attacks have surged, with reports showing that a substantial percentage of organizations have encountered smishing attempts. For instance, a report by Proofpoint highlighted that 81% of surveyed organizations in the U.S. faced such attacks in 2020, underscoring the severity of the threat [4, 5, 6]. The nature of smishing attacks often involves sending deceptive messages that prompt users to click on malicious links or provide sensitive information, making it a particularly insidious form of phishing [7, 8].

A. Contribution

The contribution of this paper is a novel, AI-driven approach for smishing detection in Android devices, utilizing TinyBERT for effective feature extraction and Aquila Optimization (AO) for hyperparameter tuning. This method enhances detection accuracy while reducing false positives by leveraging optimized

deep learning techniques. Comparative analysis shows that the proposed model outperforms traditional machine learning and deep learning models, offering a robust solution for mobile cybersecurity against SMS phishing attacks.

B. Organization

The organization of this paper is as follows: Section II presents a comprehensive review of related work. Section III details the proposed approach. Section IV outlines the experimental setup and performance metrics used to evaluate the model. Finally, Section V concludes the paper.

II. RELATED WORK

Akande et al. [9] proposed a rule-based mobile application for detecting and preventing smishing attacks by intercepting incoming SMS and forwarding them through an API to a rule-based machine learning model for classification. Gupta et al. [10] proposes a hybrid deep learning model leveraging feature selection and Ant Lion Optimization for hyperparameter tuning, achieving a detection accuracy of 97% for IoT cyberattacks. Zhou et al. [11] introduces a deep learning-based phishing detection model for Semantic Web systems, leveraging MobileBERT for feature extraction and covariance matrix adaptation evolution strategy (CMA-ES) for hyperparameter optimization, achieving a classification accuracy of 95%.

Joo et al. [12] proposed an enhanced security model called “S-Detector” that utilizes a Naive Bayes classifier for detecting smishing attacks in mobile computing environments. By filtering SMS messages through statistical learning, the model distinguishes between normal and smishing messages, effectively addressing security concerns in mobile environments. Gupta et al. [13] proposes a novel phishing detection model that integrates Random Forest for feature selection and the Brown-Bear Optimization Algorithm (BBOA) for hyperparameter tuning of a CNN, achieving 93% accuracy and 92% precision. Gaurav et al. [14] proposes an AI-driven Semantic Web phishing detection framework for e-commerce systems, integrating Chi-square for feature selection and the JADE algorithm for hyperparameter optimization of a CNN, achieving 93% accuracy with consistent stability across epochs.

Mehmood et al. [15] proposed a deep learning approach combining Convolutional Neural Network (CNN) and Long

*Corresponding Author

Short-Term Memory (LSTM) networks to detect smishing attacks, focusing on reducing false-positive predictions. This hybrid CNN-LSTM model leverages deep learning's feature extraction capabilities to improve classification accuracy and reduce incorrect classifications of legitimate messages as malicious. Gupta et al. [16] introduces a lightweight deep-learning model for phishing detection in IoT, incorporating a two-fold optimization approach with ANOVA F-test for feature selection and the Cuckoo Search algorithm for hyperparameter tuning, achieving a phishing detection accuracy of 91%.

Mishra and Soni [17] proposed a two-phase smishing detection model for smartphones, focusing on evaluating the legitimacy of URLs within SMS messages and classifying message content based on extracted features. The model includes a Domain Checking Phase to verify URLs and an SMS Classification Phase that uses the Backpropagation Algorithm for message classification. Gupta et al. [18] introduces a novel phishing detection framework by integrating RNN for sequential data processing with Whale Optimization Algorithm (WOA) for hyperparameter tuning, achieving an accuracy of 92% on a URL-based dataset.

Gowri and Brindha [19] proposed an innovative smishing detection model using Graph Neural Networks (GNNs) and multi-modal classification, incorporating both textual and image data for comprehensive feature analysis. The approach includes domain identification to evaluate sender reputation and GNN-based classification, effectively capturing relational patterns in message-domain interactions. Vajrobol et al. [20] analyzes phishing URLs using mutual information for feature selection and logistic regression for classification, achieving an exceptionally high accuracy of 99.97%, and identifies key features for distinguishing phishing attempts.

III. PROPOSED WORK

Our proposed smishing detection method (Figure 1) first gathered labeled valid and spam texts from a publicly available Kaggle SMS dataset. We cleaned, standardized the data by removing special characters, digits, and punctuation, translating text to lowercase, eliminating stopwords, and applying lemmatization in the step of data preprocessing. Using a lightweight transformer model, TinyBERT, we extracted contextual embeddings of SMS messages. T-SNE visualization of these embeddings confirmed good feature separation. By balancing exploration and exploitation in the hyperparameter search space, the Aquila Optimization (AO) method helped us to finally maximize the hyperparameters of our deep learning model, hence producing excellent smishing detection results.

Our proposed smishing detection approach follows a structured pipeline from data preprocessing to hyperparameter optimization. Below, we present the mathematical model for each stage.

A. Data Collection and Preprocessing

Given the raw SMS dataset $D = \{(x_i, y_i)\}_{i=1}^N$, where x_i represents an individual SMS message, and $y_i \in \{0, 1\}$ indi-

cates the label (0 for legitimate, 1 for spam). The preprocessing stage transforms each x_i through a series of steps:

- **Text Cleaning:** Remove special characters, numbers, and punctuation.
- **Lowercasing:** Convert x_i to lowercase:

$$x'_i = \text{lowercase}(x_i) \quad (1)$$

- **Stopword Removal:** Remove common stopwords from x'_i .
- **Lemmatization:** Reduce words to their base forms.

Let $T(x_i)$ denote the complete preprocessing function applied to x_i :

$$x''_i = T(x_i) \quad (2)$$

where x''_i is the processed text ready for feature extraction.

B. Feature Extraction using TinyBERT

The preprocessed messages x''_i are passed through TinyBERT to obtain embeddings. Let f_{TinyBERT} represent the TinyBERT model, which maps each x''_i to a feature vector v_i in the embedding space:

$$v_i = f_{\text{TinyBERT}}(x''_i) \quad \text{where } v_i \in \mathbb{R}^d \quad (3)$$

Here, d is the dimensionality of the TinyBERT embeddings. These embeddings serve as the input features for the smishing detection model.

C. Hyperparameter Optimization using Aquila Optimization (AO)

Let θ represent the set of hyperparameters for the deep learning model. AO searches for the optimal set θ^* that minimizes the objective function $\mathcal{L}(\theta)$, typically the classification error or loss function on a validation set.

- **Initialization:** Initialize a population of candidate hyperparameters $\{\theta_k\}_{k=1}^K$.
- **Objective Evaluation:** For each candidate θ_k , calculate the loss $\mathcal{L}(\theta_k)$.
- **Update Rule:** Update each candidate θ_k based on AO's exploration and exploitation strategies to find a better solution in the hyperparameter space.

The optimization process iterates until convergence, resulting in the optimal hyperparameters:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta) \quad (4)$$

D. Model Training and Prediction

Using the optimal hyperparameters θ^* , the model $f_{\text{model}}(v_i; \theta^*)$ is trained on the TinyBERT embeddings $\{v_i\}$. The model learns to classify each SMS as legitimate or spam:

$$\hat{y}_i = f_{\text{model}}(v_i; \theta^*) \quad (5)$$

where $\hat{y}_i \in \{0, 1\}$ is the predicted label for each SMS.

This mathematical flow captures each stage, from data collection to model training, culminating in a robust smishing detection system.

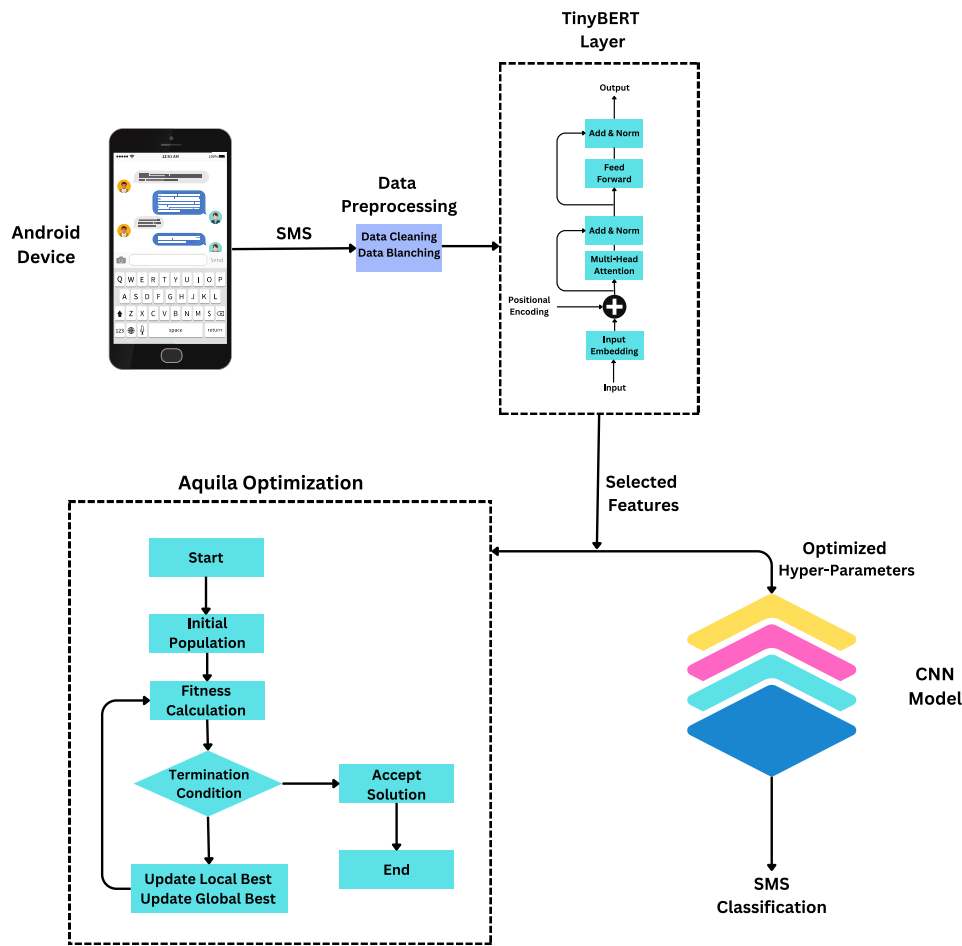


Fig. 1: Proposed Model

IV. RESULT AND DISCUSSION

A. Dataset Representation

Widely utilized for SMS spam detection research, the SMS Spam Collection dataset used for this study came from Kaggle[21]. There are 5,574 English SMS messages altogether, tagged as either "ham," (legal) or "spam." Two columns—"v1" for the label and "v2" for the SMS contents—represent every communication. The dataset combines messages coming from many sources: 425 spam messages extracted from the Grumbletext website, a UK-based forum where users report SMS spam; 3,450 legitimate messages from the NUS SMS Corpus, mostly contributed by students at the National University of Singapore; 450 legitimate messages sourced from Caroline Tag's PhD thesis; and 1,424 messages (1,002 legitimate and 322 spam) from the SMS Spam Corpus v.0.1 Big, previously used in academic research. Training and testing smishing detection algorithms can be based on this extensive collection of SMS data with dependability.

B. Dataset Preprocessing

We carried out many steps to standardize and clean the SMS text data in the stage of dataset preparation. To rid any

non-alphabetic symbols that would not add to the meaning of the communications, we first deleted special characters, digits, and punctuation. We then turned all of the text lowercase to preserve consistency and prevent word duplication across several contexts. Common stopwords (like "the," "is," and "and") that do not significantly provide semantic meaning for our study were then deleted. At last, we lemmatized—that is, stemmed—reducing words to their root forms, so aggregating related terms (e.g., "running" and "runs" become "run"). This increases the meaning of the data.

Following preprocessing, we created a word cloud (Figure 2) to show the most often occurring terms in the SMS corpus. Prominent in both valid (ham) and spam messages, the word cloud exposes often used words such "call," "get," "go," "free," and "need." This graphic shows phrases that might be especially important for differentiating smishing (spam) from valid communications and offers understanding of the language patterns found in SMS data.

C. Feature Extraction Using TinyBERT

We produced embeddings from the SMS texts using TinyBERT in the feature extraction phase. Essential for differentiating between smishing (spam) and legal (ham) messages,



Fig. 2: Word Cloud Representation

TinyBERT, a lightweight and efficient variation of the BERT paradigm, detects contextual elements of the text. We project the high-dimensional TinyBERT embeddings into a 2D space using t-Distributed Stochastic Neighbour Embedding (t-SNE), a dimensionality reduction method, so visually evaluating the separability of these features.

Each point in the resulting t-SNE visualization (Figure 3) denotes each SMS message, hence displaying clusters of data points. The colors signal the message labels; yellow denotes spam and purple denotes ham. The division of spam and ham clusters in some regions implies that the TinyBERT embeddings effectively capture important variations between the two classes, therefore offering a strong basis for our smishing detection system. This animation shows how well TinyBERT codes elements that might distinguish smishing efforts from official SMS material.

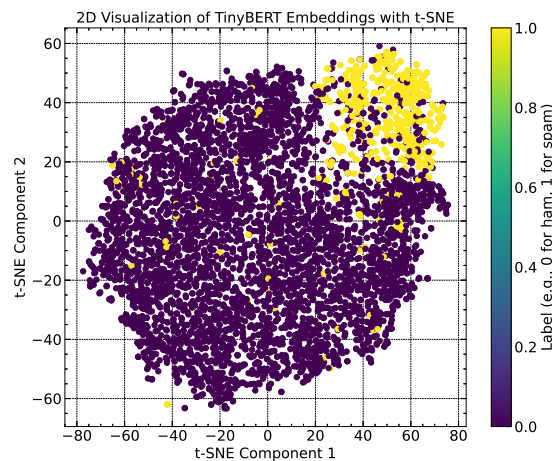


Fig. 3: Performance of TinyBERT

D. Performance of Aquila Optimization (AO)

Following TinyBERT feature extraction, we constructed a deep learning model tuned with Aquila Optimization (AO) method to identify the best hyperparameters. The next charts show AO's performance all through the optimization process.

- The runtime chart (Figure 4a) displaying in seconds the time taken for every iteration of AO. The differences in runtime show the computing load AO needs to investigate and use various hyperparameter settings. The

runtime stays rather constant, with small variations as the method tries to optimally balance stages of exploration and exploitation.

- Diversity Measurement Chart (Figure 4b) Tracking the variety of solutions over iterations, the diversity measurement chart offers understanding of the search space exploration. The variety is first more reflective of the exploratory phase of the algorithm. Diversity reduces as AO advances, suggesting a concentration on exploitation; AO then narrows down on likely areas of the solution space to hone the ideal parameters.
- Percentage of Exploration vs. Exploitation (Figure 4c) chart shows how to strike the balance between exploitation and discovery. AO gives exploration top priority early in the optimization process in order to cover a large spectrum of feasible solutions. As the algorithm converges, the focus moves to exploitation, guaranteeing that the best solutions are polished further for highest accuracy in hyperparameter choice.
- Global Objectives Chart (Figure 4d) Across iterations, this chart shows the objective function values. The declining trend shows AO's achievement in reducing the objective function, thereby obtaining ever better hyperparameter values. The stabilization at the end indicates that AO has most likely attained ideal or almost ideal conditions for the deep learning model.

E. Performance of Deep Learning Model

Following Aquila Optimization (AO) to find the best hyperparameters, we trained the deep learning model for five epochs to preserve a lightweight and effective structure fit for Android devices. A confusion matrix and a classification report helped one assess the performance of the model.

Figure 5 presents the confusion matrix. The model appropriately found 119 spam messages and 882 valid (ham) messages in this matrix. Still, there were 26 false negatives—spam mails misclassified as legitimate—and 7 false positives—legitimate messages misclassified as spam. With appropriate performance for spam detection as well, the great number of accurate classifications—especially in the genuine class—indicate the model's robustness in spotting ham messages.

Classification Report For every class is presented in Figure 6, the classification report offers comprehensive measures including F1-score, recall, and precision. The model minimizes false positives in both classifications rather well: 97% for authentic messages and 94% for spam. At 82%, the recall for spam messages was rather lower, suggesting some space for development in terms of all spam occurrence capture. With an F1-score of 0.98 for valid communications and 0.88 for spam, the model's overall accuracy—which shows a strong mix between precision and recall—was 97%.

F. Comparative Analysis

We performed a comparison study against conventional machine learning and deep learning models including GRU, LSTM, RNN, Logistic Regression, Simple Neural Network,

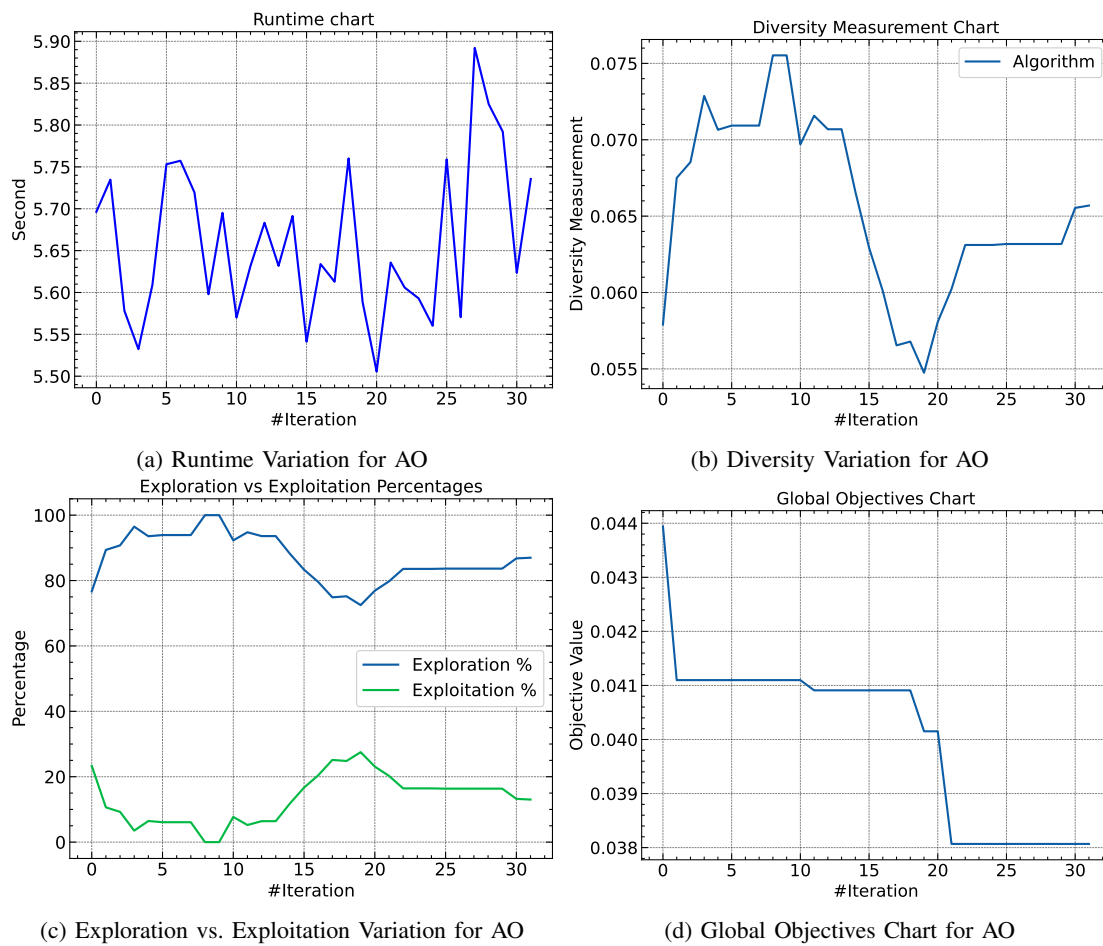


Fig. 4: Performance of AO

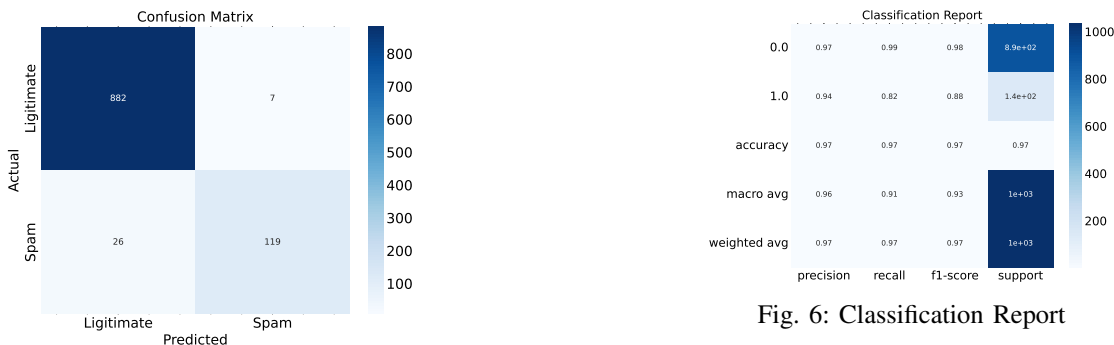


Fig. 5: Confusion Matrix

Fig. 6: Classification Report

and SVM to evaluate the efficacy of our proposed model. While the accuracy and loss variation during training are shown in line graphs, the assessment metrics—which are shown in the bar charts—accuracy, precision, recall, and F1 score.

- **Accuracy and Loss Comparison (Figures 7a and 7b):** Over the course of the training, the line graphs show the accuracy and loss of every model. Compared to other models, our suggested model attained a lower loss and

a better accuracy, which helped to stabilize fast over the training phase. This shows for the suggested model effective convergence and a strong training process.

- **Comparative Accuracy (Figure 7c):** The accuracy numbers for every model show that our suggested model attained an accuracy of 96.81%, which is equivalent to SVM (96.61%), and much higher than other models (all obtaining roughly 85.98%).
- **Figure 7d** presents precision comparison whereby our suggested model attained a precision of 96.76%, closely followed by SVM with 96.61%. Other models display a

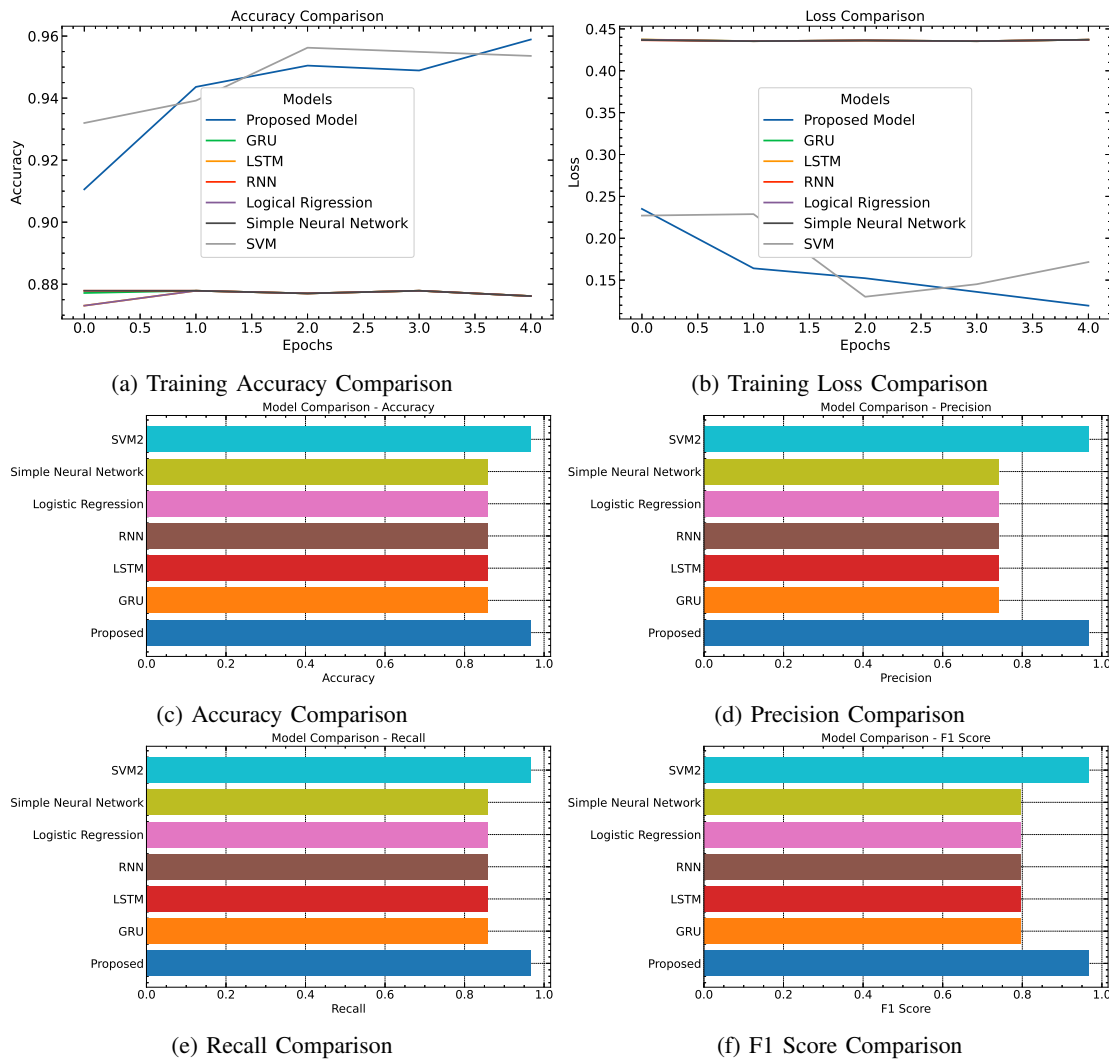


Fig. 7: Comparative Analysis

less than 73.92% precision.

- Recall Comparison (Figure 7e): While other models scored roughly 85.98%, the suggested model reached 96.81% recall, same to SVM at 96.61
- F1 Score Comparison (Figure 7f): The comparison of the F1 scores shows the harmony between recall and accuracy. With an F1 score of 96.71%, our suggested model somewhat exceeded SVM (96.61%), and far above other models (around 79.49)

These findings show that whilst outperforming other standard models like GRU, LSTM, and RNN, our proposed model performs at par with or better than SVM across all evaluation measures. This emphasizes how well our adjusted deep learning model detects smishing on Android devices.

V. CONCLUSION

In this study, we proposed an AI-driven smishing detection model for Android devices, leveraging TinyBERT for feature extraction and Aquila Optimization (AO) for hyperparameter

tuning. The model was trained on a publicly available SMS dataset from Kaggle, with rigorous preprocessing to improve message clarity. Visualization of TinyBERT embeddings using t-SNE demonstrated clear feature separability between legitimate and spam messages. The optimized model achieved superior performance, with 96.81% accuracy, 96.76% precision, 96.81% recall, and a 96.71% F1 score, outperforming standard models like GRU, LSTM, and SVM. These results underscore the model's robustness and efficiency in detecting smishing, providing a highly accurate solution for enhancing mobile security against SMS phishing attacks in Android environments.

ACKNOWLEDGMENT

This research work is supported by National Science and Technology Council (NSTC), Taiwan Grant No. NSTC112-2221-E-468-008-MY3.

REFERENCES

- [1] P. Shah and A. Agarwal, "Cybersecurity behaviour of smartphone users in india: an empirical analysis," *Information and Computer Security*, vol. 28, pp. 293–318, 2020.
- [2] D. Odera, "A comparative analysis of recurrent neural network and support vector machine for binary classification of spam short message service," *World Journal of Advanced Engineering Technology and Sciences*, vol. 9, pp. 127–152, 2023.
- [3] A. Ishtaiwi, A. M. Ali, A. Al-Qerem, M. Sabahean, B. Alzubi, A. Almomani, M. Alauthman, A. Aldweesh, and M. A. Al Khaldy, "Next-gen phishing defense enhancing detection with machine learning and expert whitelisting/blacklisting," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 14, no. 1, pp. 1–17, 2024.
- [4] M. Clasen, F. Li, and D. Williams, "Friend or foe: an investigation into recipient identification of sms-based phishing," pp. 148–163, 2021.
- [5] C. Surianarayanan and P. R. Chelliah, "Integration of the internet of things and cloud: Security challenges and solutions—a review," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 13, no. 1, pp. 1–30, 2023.
- [6] T. F. D. B. Sarasola, A. García, and J. L. Ferrando, "Iiot protocols for edge/fog and cloud computing in industrial ai: A high frequency perspective," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 14, no. 1, pp. 1–30, 2024.
- [7] H. Wahsheh and M. Al-Zahrani, "Lightweight cryptographic and artificial intelligence models for anti-smishing," pp. 483–496, 2021.
- [8] S. Gaber and M. Alenezi, "Transforming application development with serverless computing," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 14, no. 1, pp. 1–16, 2024.
- [9] O. N. Akande, O. Gbenle, O. C. Abikoye, R. G. Jimoh, H. B. Akande, A. O. Balogun, and A. Fatokun, "Smsprotect: An automatic smishing detection mobile application," *ICT Express*, vol. 9, no. 2, pp. 168–176, 2023.
- [10] B. B. Gupta, A. Gaurav, R. W. Attar, V. Arya, S. Bansal, A. Alhomoud, and K. T. Chui, "A hybrid ant lion optimization algorithm based lightweight deep learning framework for cyber attack detection in iot environment," *Computers and Electrical Engineering*, vol. 122, p. 109944, 2025.
- [11] L. Zhou, A. Gaurav, V. Arya, R. W. Attar, S. Bansal, and A. Alhomoud, "Enhancing phishing detection in semantic web systems using optimized deep learning models," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 20, no. 1, pp. 1–13, 2024.
- [12] J. W. Joo, S. Y. Moon, S. Singh, and J. H. Park, "S-detector: an enhanced security model for detecting smishing attack for mobile computing," *Telecommunication Systems*, vol. 66, no. 1, pp. 29–38, Sep. 2017.
- [13] B. B. Gupta, A. Gaurav, R. W. Attar, V. Arya, S. Bansal, A. Alhomoud, and K. T. Chui, "A hybrid cnn-brown-bear optimization framework for enhanced detection of url phishing attacks," *Computers, Materials & Continua*, vol. 81, no. 3, 2024.
- [14] A. Gaurav, K. T. Chui, V. Arya, R. W. Attar, S. Bansal, A. Alhomoud, and K. Psannis, "Optimized ai-driven semantic web approach for enhancing phishing detection in e-commerce platforms," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 20, no. 1, pp. 1–13, 2024.
- [15] M. K. Mehmood, H. Arshad, M. Alawida, and A. Mehmood, "Enhancing smishing detection: A deep learning approach for improved accuracy and reduced false positives," *IEEE Access*, 2024.
- [16] B. B. Gupta, A. Gaurav, V. Arya, R. W. Attar, S. Bansal, A. Alhomoud, and K. Tai Chui, "Cuckoo search-optimized deep cnn for enhanced cyber security in iot networks," *Computers, Materials & Continua*, vol. 81, no. 3, 2024.
- [17] S. Mishra and D. Soni, "Dsmishsms-a system to detect smishing sms," *Neural Computing and Applications*, vol. 35, no. 7, pp. 4975–4992, Mar. 2023.
- [18] B. B. Gupta, A. Gaurav, R. W. Attar, V. Arya, A. Alhomoud, and K. T. Chui, "Optimized phishing detection with recurrent neural network and whale optimizer algorithm," *Computers, Materials & Continua*, vol. 80, no. 3, 2024.
- [19] K. Gowri and S. Brindha, "Advancements in smishing detection: A novel approach utilizing graph neural networks," *COMPUTER*, vol. 24, no. 7, 2024.
- [20] V. Vajrobol, B. B. Gupta, and A. Gaurav, "Mutual information based logistic regression for phishing url detection," *Cyber Security and Applications*, vol. 2, p. 100044, 2024.
- [21] T. Almeida and J. Hidalgo, "SMS Spam Collection," UCI Machine Learning Repository, 2011, DOI: <https://doi.org/10.24432/C5CC84>.