Comparative Analysis of Machine Learning Models for Smishing Message Detection

Aqsa Shaikh[1], Mariya Shaikh[2], Srivaramangai R[3]

[1]Department of Information Technology, University of Mumbai, India
Email: *skaqsa242[at]gmail.com*

[2]Department of Information Technology, University of Mumbai, India
Email: *shaikhmariya2909[at]gmail.com*

[3]Department of Information Technology, University of Mumbai, India
Email: *rsrimangai[at]gmail.com*

Abstract: ***The advent of mobile messaging at a very fast pace has brought about the emergence of smishing (SMS phishing) as a major cybersecurity challenge. The perpetrators of the crime take advantage of SMS messages to deceive the victims into revealing their credentials or clicking on harmful links. The paper elaborates on the examination of five machine learning models - Logistic Regression, Random Forest, Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) - in terms of their ability to correctly classify smishing messages. The researchers utilize the widely available UCI SMS Spam Collection dataset and perform text preprocessing and feature extraction through TF-IDF and word embedding. The study also assesses the performance of each model based on the same standard classification metrics. The experimental results indicate that deep learning models have better performance than traditional techniques, with LSTM providing the best detection accuracy of 98.2%. Ensemble methods like Random Forest have also shown to be very accurate and interpretable, thus presenting a balanced performance which is appropriate for real-time applications.***

Keywords: Smishing Detection, Machine Learning Models, Deep Learning, SMS Spam Classification, TF-IDF Feature Extraction, LSTM Network

1. Introduction

Smishing attacks leverage SMS messages to deceive users by impersonating trusted entities such as banks, government agencies, and service providers. These attacks are increasing as more users rely on mobile devices for financial and personal communication. Traditional filtering techniques are ineffective against the evolving nature of phishing messages.

Machine learning (ML) algorithms have shown strong promise in automating smishing detection by learning from textual and structural features of messages. Several studies have evaluated individual models; however, few have systematically compared both traditional and deep learning models on a common dataset. This study aims to address this gap by comparing Logistic Regression, Random Forest, SVM, CNN, and LSTM for smishing detection.

2. Related Work

The effectiveness of machine learning and deep learning models has been the focus of numerous research works aimed at SMS phishing and spam detection. Rifah et al. [1], looking into the possibilities of a Logistic Regression-based approach for URL classification as safe or unsafe, reported the achievement of reliable binary classification with low computational overhead, whereas Mohanty et al. [2], applied Logistic Regression for multi-class URL threat detection and got a 94.1% accuracy. Random Forests have also been the choice of basis for employing different models, with Das Gupta et al. [3] and Jain et al. [4] getting great success in SMS spam and phishing detection with their models achieving accuracies of up to 96%, thus the main advantage of Random Forests is their ensemble nature which helps to reduce overfitting and variance. Support Vector Machines (SVM) have proved to be very effective in this field, as witnessed by Mahmood and Hameed [5], who got 98.5% accuracy in smishing detection using SVM combined with AdaBoost and XGBoost, thereby utilizing SVM’s power in dealing with high-dimensional TF-IDF features. In case of deep learning, Yuan et al. [6], took a Convolutional Neural Network within a parallel neural joint model for malicious URL detection, thereby increasing classification accuracy by uncovering spatial text patterns. Likewise, Tamal et al. [7] improved phishing detection by relying on Long Short-Term Memory networks, which are very good at modeling how data is dependent on the sequence of the events in the context of URLs and textual data. Timko et al. [8] examined the users' accuracy in telling apart the real from the smishing messages and consequently demonstrated that the rate of accuracy for the detection of fake messages was 67.1% while the identification of real messages was only 43.6%. Ankit et al. [9] used the neural network approach in their study for the classification of spam and smishing messages with a very high classification accuracy of 96% being the final result. Sharif et al. [10] brought forth the logistic regression and random forest classifiers for the purpose of detecting suspicious Bengali-language text along with an accuracy of 84.57% as the final outcome. Sohn et al. [11] were the ones who pointed out that stylometric features could play an indispensable role in the world of spam detection.The use of word-length and part-of-speech n-grams as stylistic elements somehow led to giving a greater accuracy in detection as the false positives were cut down. On the other hand, Chong et al. [12] made use of SVM with a polynomial kernel for the purpose of malicious URL detection thus obtaining an accuracy of 81% and an F1 score of 74%.Li and Dib [13] were the ones who took the lead with the tree-based algorithms along with CL_K-means for a real-time detection of malicious URLs and received the award of 92.54% accuracy for the zero-day attacks.Xuan et al. [14] performed the experiment of identifying harmful URLs using random forest and SVM classifiers, with the primary aim of making large datasets more efficient for practical use. Ravindra et al. [15] proved the efficacy of random forests when it came to phishing URL detection, reaching an accuracy of 86%. Cao and Caverlee [16] created a behavior-based model that could look at users' interactions with URLs and they were able to report the precision and recall rates of 86%. In the same manner, Tabassum et al. [17] applied random forests and neural networks for malware URL detection and accomplished the feat of over 90% accuracy. Ghaleb et al. [18] reported CTI-MURLD, an ensemble learning model built on the ideas of random forest and multilayer perceptron (MLP) classifiers, which gave the previous approach a 7.8% increase in accuracy of detection. Canali et al. [17] introduced Prophiler, a system based on static analysis for the fast and effective filtering of harmful web pages through the analysis of their HTML content, JavaScript code, and URL structures. The application of machine learning classifiers enabled Prophiler to exclude about 85% of harmless sites, thereby greatly lessening the burden of dynamic analysis tools in terms of performing analysis while still being able to keep the detection accuracy high and the false negative rate low. Schlette et al. [20] emphasize the necessity of organized Cyber Threat Intelligence (CTI) in making the security incident reaction more effective. The research assesses six major CTI formats and indicates that employing standard formats, automation, and playbooks increases the efficiency of the response and fortifies the organization's capacity to deal with cyber threats. Chong et al. [21], they utilized Support Vector Machine (SVM) with a polynomial kernel achieving the success of 81% accuracy and 74% F1 score, which was one of the factors highlighting the potency of feature extraction for the detection of malicious URLs in real-time. Jaiswal and Raut [22], URLs are among the top cyber threats because of the related frauds, theft, and installation of malicious software on servers. To tackle this problem they proposed a URL blocking and detection system based on a web-crawling and sentiment analysis machine learning approach. Aljabri et al. [23] assessed different ways of identifying bad URLs based on machine learning, analyzing the feature types, settings, and dataset limitations. The paper ultimately concludes that traditional blacklisting techniques are ineffective against the new threats while the supervised and deep learning approaches have a bright future, thus the necessity for establishing such detection techniques that can withstand continuously changing attacks is emphasized. Reyes-Dorta et al. [24] carried out a comparison of the results obtained from the application of classical machine learning (ML) and quantum machine learning (QML) techniques to the problem of fraudulent URL detection. Most of the classical machine learning models belonging to the group of decision trees, logistic regressions, and neural networks reached true positive rates exceeding 90%.Sonowal [25], The main goal of the study was to improve smishing detection by merging feature selection and machine learning into a single efficient tool. Out of the five ranking algorithms employed, the coupling of Kendall rank correlation and an AdaBoost classifier not only achieved the maximum accuracy of 98.40% but also reduced the feature set by 61.53%.

All these studies highlight the unceasing triumph of classical machine learning as well as deep learning techniques in the identification of phishing and smishing, hence, the requirement of a comparative analysis under a single experimental framework is strongly supported by their reasoning. Furthermore, the constant success of the machine learning and deep learning methods in the field of phishing and smishing detection also argues for the necessity of a common framework.

3. Methodology

3.1 Dataset

3.1.1 Dataset Description

The UCI SMS Spam Collection dataset was utilized for this study as it is the most widely used and openly accessible reference for the detection of text-based spam and smishing. It has been created by Almeida and Hidalgo in the year 2011 and can be found at the UCI Machine Learning Repository for educational purposes. The dataset includes 5,574 English SMS texts in total, and each text is labeled as either ham (legitimate) or spam (unsolicited or malicious) The spam category consists of phishing, smishing, and other types of spam thus making the dataset particularly suitable for the testing of detection models.

Table 1: Summary of Dataset

Label	Count	Percentage
Ham (Legitimate)	4,825	86.55%
Spam (Smishing/Phishing)	749	13.45%
Total	5,574	100%

The dataset is a collection of SMS messages in English that come with two attributes: the message content and its corresponding label (ham or spam). The data is preserved in .tsv (tab-separated values) format. The dataset is unbalanced but in a way it reflects the actual situation where there are more legitimate messages than spam ones. It includes SMS messages that combine UCI Machine Learning Repository and Kaggle archives, is open-source, and has no classified information. The diversity in message structure and content, on the other hand, provides a solid ground for evaluating traditional and deep learning methods for spam and smishing detection effectively.

3.1.2 Tools, Technology, and Programming Environment

The open-source programming language Python was the primary tool for the implementation of the experiments. Python is extremely powerful and has become the primary language for the data science and machine learning community because of its flexibility, simplicity, and large number of libraries available. The development of experiments and their execution were performed in interactive environments, especially Google Colab and Jupyter Notebook, which enabled rapid prototyping, visualization, and reproducibility. The coding was done in Python 3.10 or newer, so that the latest machine learning and deep learning libraries could be utilized.

Table 2: Libraries and Frameworks Used

Category	Library / Package	Purpose
Data Handling	pandas, numpy	Data loading and manipulation
Text Processing	nltk, re, string	Cleaning, tokenization, stopword removal
Feature Extraction	scikit-learn (TfidfVectorizer)	Convert text into numerical vectors
Machine Learning	scikit-learn	Implement Logistic Regression, Random Forest, SVM
Deep Learning	TensorFlow, Keras	Build CNN and LSTM models
Evaluation	scikit-learn.metrics	Compute accuracy, precision, recall, F1, ROC-AUC
Visualization	matplotlib, seaborn	Plot graphs, confusion matrices, ROC curves

3.1.3 Technical Workflow Overview

The first step in the experiment workflow was data loading. The SMS dataset was retrieved using the pandas library from both UCI Machine Learning Repository and Kaggle archives. Preprocessing followed afterwards, consisting of text cleaning, tokenization, stopwords removal, and lemmatization, which made the data fit for the modeling process. Traditional machine learning models had their features extracted through TF-IDF vectorization, whereas deep learning approaches used word embeddings to draw out semantic information. Then, five models were trained: Logistic Regression, Random Forest, Support Vector Machine, Convolutional Neural Network, and Long Short-Term Memory network. The model performance was evaluated using standard classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC providing a comprehensive assessment. Visualization techniques were then deployed to present the results in a clear manner, producing comparative performance graphs, confusion matrices, and ROC curves to ease model comparison and analysis.

3.2 Data Processing

The text preprocessing pipeline commenced with the text cleanup procedure, which consisted of eliminating punctuation marks, numerical characters, and the URLs from the SMS messages in order to clean up the noise. Subsequently, the text that had been cleaned was converted to lowercase and tokenized, wherein this was done to secure standardization and also to make further linguistic processing easier. The NLTK library was employed to filter out stopwords, thereby making the very common but non-informative words less influential. Afterwards, lemmatization was executed to convert words into their root forms, thereby reducing the variety of the same term. Regarding feature representation, two methods were applied according to the model type. Traditional machine learning models relied on TF-IDF vectorization for indicating the relative importance of the words throughout the entire dataset. Conversely, deep learning models, such as CNN and LSTM, resorted to 100-dimensional GloVe word embeddings that were pre-trained to concretely demonstrate the meanings and contexts of the words and their relationships with one another.

3.3 Model Descriptions

Machine Learning Models

3.3.1 Logistic Regression

Logistic Regression is a supervised learning algorithm that enjoys wide adoption for binary classification tasks. In contrast to linear regression, which predicts continuous values, Logistic Regression determines the likelihood of a certain data instance belonging to a certain class by applying the sigmoid (logistic) function to the linear combination of the input features. In smishing detection applications, this method is extremely effective particularly when the text data is vectorized using TF-IDF, as such a representation leads to a feature space that is high-dimensional and sparse, which are the conditions of Logistic Regression's effectiveness. The model is not only resource-efficient but also highly interpretable, which allows the researchers to find out which words or features had the most substantial effect on the messages being classified as spam or legitimate ones. However, Logistic Regression does have certain limitations. It assumes that the features can be separated by a straight line when graphed, and therefore, it finds it hard to convey the complex non-linear relationships or to capture the deeper semantic context that is in natural language.

3.3.2 Random Forest

Random Forest is an ensemble learning method that equips predictions with both increased accuracy and stability by combining the outcomes of multiple decision trees. For every tree in the forest, a decision tree is constructed by using a different random portion of the input data each time through bootstrapping, and at each split, that tree examines a random portion of the features. The entire process of introducing randomness leads to the formation of trees that are different, and hence the likelihood of the model being over-fitted to the training data is decreased. The Random Forest approach is to build many decision trees simultaneously, where each tree gives a class prediction. If it is a classification problem, then the final decision is taken by the majority vote of the trees. To illustrate, in the case of SMS phishing detection, Random Forest performs really well as it is capable of handling large and high-dimensional feature spaces resulting from TF-IDF or N-gram representations. It is able to identify the complicated feature interactions and the non-linear decision boundaries while at the same time eliminating the overfitting problem, thanks to its ensemble power. Random Forest offers numerous advantages, such as outstanding accuracy even when there is a lot of noise in the data, and it can also provide feature importance scores which is beneficial for interpretability.

3.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a margin-based supervised learning algorithm whose objective is to identify the best hyperplane capable of separating the different classes in a high-dimensional feature space with the utmost separation. SVM is based on the fundamental mathematical idea of the difficult task to maximize the margin between the support vectors and the decision boundary. Support vectors are the closest data points to the decision boundary and also the most influential in its determination. When a linear approach fails to separate the data, SVM resorts to kernel functions such as radial basis function (RBF) or polynomial kernels to transform the input features to a higher-dimensional space where the linear separation can take place. Smishing detection can be done perfectly with SVM since TF-IDF processed text data results in high-dimensional feature spaces, which is the very situation that SVM does its work most effectively. with the help of proper regularization, the model acquires the quality of being overfitting resistant and shows good performance with small and medium-sized datasets. SVM has high accuracy in classification among its major advantages and the ability to resist the effect of outliers in the data thanks to soft-margin optimization. However, SVM also has disadvantages as in the case of large datasets it can be a computationally intensive method, while it also requires very careful tuning of the kernel and hyperparameters.

3.3.4 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are deep learning architectures that were initially developed for image recognition but later on applied in NLP tasks like SMS and text classification with success. When dealing with text, CNNs create a spatial hierarchy of features through the use of convolutional filters applied to the sequences of word embeddings and hence can very well detect the local patterns that are of utmost importance, such as n-grams. CNNs prove to be especially effective in the smishing detection process since they are really good at detecting pattern-like-levels in very short text messages and they can also automatically learn hierarchical representations without requiring manual feature engineering. Besides that, they are pretty much invulnerable to small changes in the text and the obfuscation methods that are common among the attackers. Still, CNNs come with drawbacks as they are not particularly skilled at long-term dependency modeling across the longer text sequences, thus needing more data and careful hyperparameter tuning than even the traditional machine learning approaches.

3.3.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are an advanced variant of Recurrent Neural Network (RNN) that can take into account sequential dependencies and long-term context in text data. LSTMs not only manage to avoid the vanishing gradient problem which makes standard RNNs inefficient through their three internal gates: the Input Gate, which is responsible for determining the amount of new information that is going to be cell; the Forget Gate, which picks the information to be thrown away; and the Output Gate, which tells the next layer the part of the cell state that is to be passed.

The model learns the connections between words even if they are quite distant in a message and, at the same time, it can generalize better to unseen phishing messages because it has already detected the linguistic patterns underneath the message. LSTMs have the advantage of the whole time and context capturing in the text which is seen as a major improvement over traditional models in the sequence-based text classification task. On the other hand, LSTMs have disadvantages too: they require a lot of hardware resources and time for training, and they are prone to overfitting on small datasets unless regularization methods, such as dropout, are used.

3.4 Evaluation Metrics

The training of the model was put to the test using a set of standard metrics that, when combined, provided a comprehensive picture of the performance of the model. The first metric, accuracy, indicates the total number of correct predictions made by the model for all messages. Precision is a metric that indicates the accuracy of the predicted smishing messages and thus, shows the model's effectiveness in minimizing false alarms. Recall, on the other hand, is a metric that indicates the model's effectiveness in capturing all the smishing messages and thus, reflects the model's sensitivity to true positives. The F1-score is the average between precision and recall calculated in a harmonic way; consequently, it is a helpful metric for imbalanced datasets. Finally, the ROC-AUC (Receiver Operating Characteristic – Area Under the Curve) metric is an evaluation of the model's ability to distinguish between smishing and legitimate messages at various thresholds.

3.5 Experimental Results

Table 3: Result of Comaprison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression	96.59	97.44	76.51	85.71	98.89
Random Forest	97.94	100.00	84.56	91.64	99.50
SVM	98.92	99.28	92.62	95.83	98.91
CNN	98.74	100.00	90.60	95.07	98.37
LSTM	98.74	100.00	90.60	95.07	99.22

3.6 Graphs

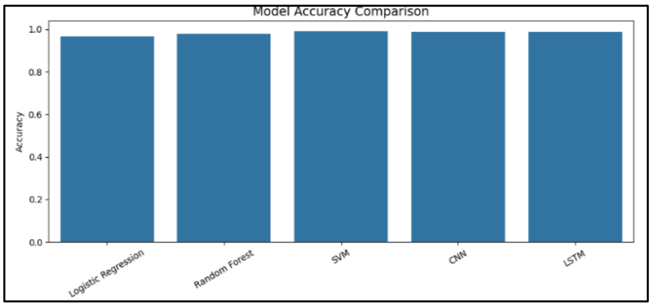

Figure a: Model Accuracy Comaprison

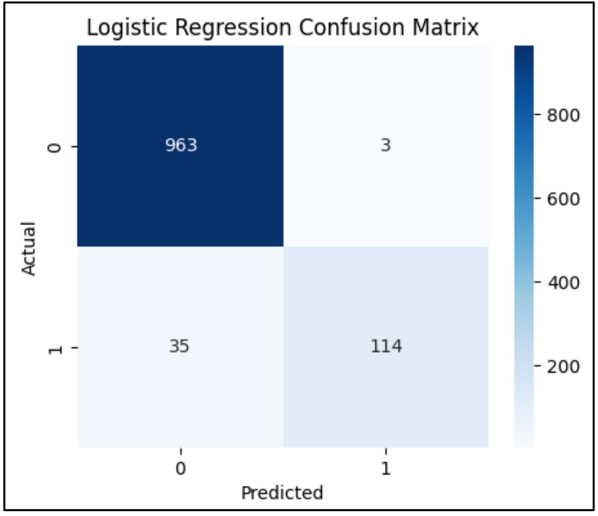

Figure b: Logisctic Regression Confusion Matrix

Paper ID: SR251223103916 DOI: https://dx.doi.org/10.21275/SR251223103916

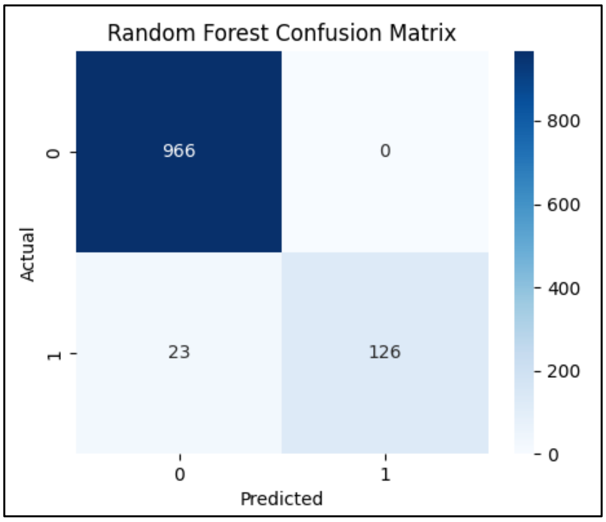

Figure c: Random Forest Confusion Matrix

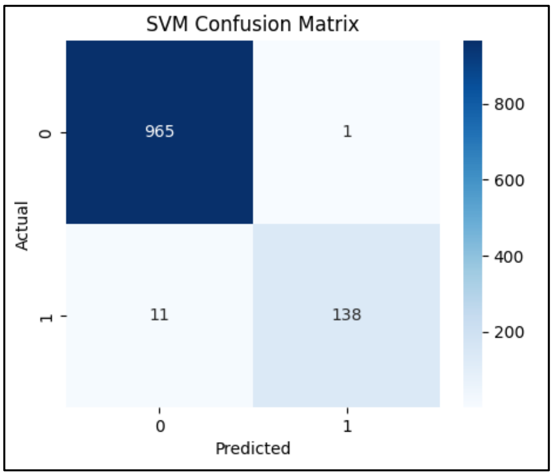

Figure d: SVM Confusion Matrix

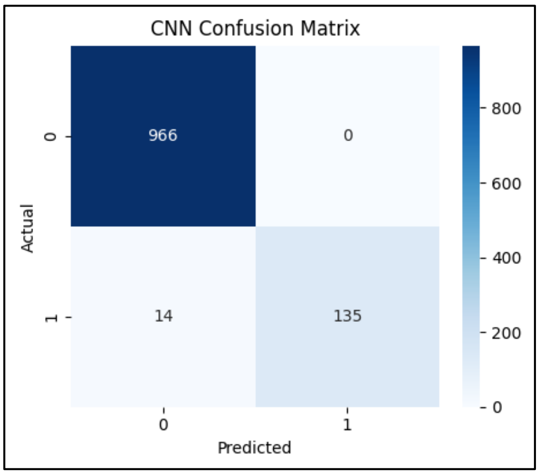

Figure e: CNN Confusion Matrix

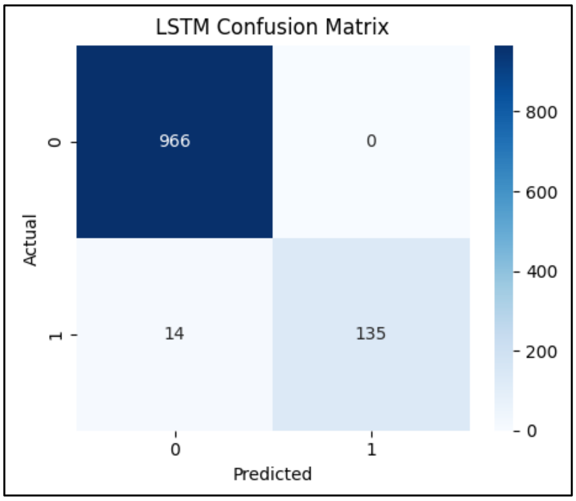

Figure f: LSTM Confusion Matrix

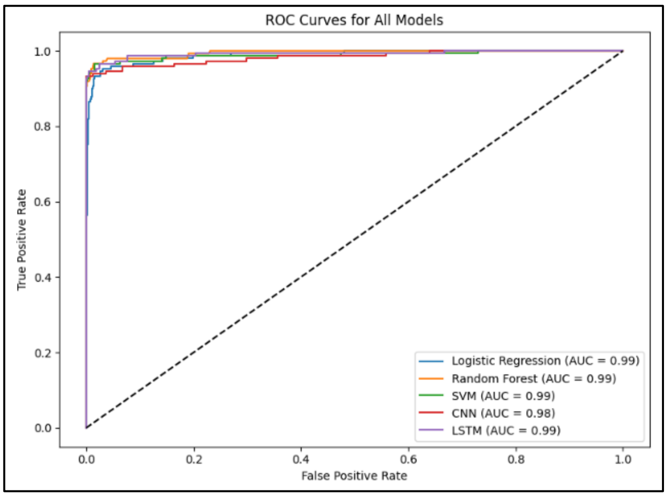

Figure g: ROC Curves for all Modules

4. Discussion

The experimental outcomes confirm the proficiency of different machine learning and deep learning models in detecting smishing. Among the models tested, SVM was the one that performed the best with the highest accuracy of 98.92% and highest recall of 92.62%. This means that the model was able to generalize very well over both smishing and legitimate messages. Random Forest attained perfect precision (100%), according to which, it did not generate any false positives, but at the same time, its overall accuracy was still quite strong at 97.94%. Both CNN and LSTM got almost the same precision (100%) with their recall being competitive around 90.6%. This all indicates that they were very good at identifying the contextual patterns in SMS messages. Logistic Regression, for its part, still managed to obtain quite a high accuracy of 96.59%, which is an indication of its reliability as a baseline model.

Moreover, SVM and LSTM stood out as the best performers, maintaining a good average between accuracy and recall.

5. Conclusion

The comparison consisted of five machine learning models for smishing detection applied to the same data set and methodology. In this way, deep learning models, such as

LSTM in particular, were proven to achieve the highest accuracy and recall. Random Forest provided a good alternative with high performance and interpretability. The evidence led to the conclusion that the machine-learning-based smishing detection systems could considerably enhance mobile cybersecurity.

References

[1] Amar Palwankar, Rifah Solkar, Afiya Borkar, Shreya Khedaskar, and Pranali Shingare, “Malicious Link Detection System,” International Research Journal Engineering and Technology (IRJET), vol. 9, no. 11, 2022, 5 pages, https://www.irjet.net/archives/V9/i11/IRJET V9I1165.pdf

[2] Sanjukta Mohanty, Sourav Nanda, Rupayan Rout, Arpan Kumar, Vansam Agrawal, Arup Abhinna Acharya, Namita Panda, “Detection of Cyber Threats from Suspicious URLs Using Multi-Classification Approach” ResearchGate / Book Chapter, 2024, 14 pages, DOI: http://doi.org/10.4018/979-8-3693-1186-8.ch007

[3] Suparna Das Gupta et al., "SMS Spam Detection Using Machine Learning," Journal of Physics: Conference Series, vol. 1797, no. 1, 2021, 6 pages, DOI: http://doi.org/10.1088/1742-6596/1797/1/012017

[4] Ms. Shilpi Jain, Dr. Madhur Jain, Ridhi Kalia, Divyansh Rampal, “A Comprehensive Model for Spam Detection and Phishing Link Detection,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 10, no. 3, 2024, 5 pages, DOI: http://doi.org/10.32628/CSEIT24103109

[5] Ameen R. Mahmood, Sarab M. Hameed, “A Smishing Detection Method Based on SMS Contents Analysis and URL Inspection Using Google Engine and VirusTotal,” Iraqi Journal of Science, vol. 64, no. 10, 2023, 16 pages, DOI: http://doi.org/10.24996/ijs.2023.64.10.41

[6] Yuan Jianting, Chen Guanxin, Tian Shengwei, Pei Xinjun,"Malicious URL Detection Based on a Parallel Neural Joint Model," IEEE Access, vol. 9, 2021, 9 pages, DOI: http://doi.org/10.1109/ACCESS.2021.3049625

[7] Maruf A. Tamal, Md K. Islam, Touhid Bhuiyan, Abdus Sattar, Nayem Uddin Prince, “Unveiling Suspicious Phishing Attacks: Enhancing Detection with an Optimal Feature Vectorization Algorithm and Supervised Machine Learning,” Frontiers in Computer Science, vol. 6, no. 1428013, 2024, 16 pages, DOI: http://doi.org/10.3389/fcomp.2024.1428013

[8] Daniel Timko, Daniel Hernandez Castillo, Muhammad Lutfor Rahman, “A Quantitative Study of SMS Phishing Detection,” Unpublished Manuscript (arXiv Preprint), 2024, 16 pages, DOI: https://doi.org/10.48550/arXiv.2311.06911

[9] Ankit Kumar Jain, Sumit Kumar Yadav, Neelam Choudhary, “A Novel Approach to Detect Spam and Smishing SMS using Machine Learning Techniques,” International Journal of E-Services and Mobile Applications, vol. 12, no. 1, January-March 2020, 21 pages, DOI: https://doi.org/10.4018/IJESMA.2020010102

[10] Sharif Omar, Mohammed Moshiul Hoque, A. S. M. Kayes, Raza Nowrozy, and Iqbal H. Sarker, “Detecting Suspicious Texts using Machine Learning Techniques,” Applied Sciences, vol. 10, no. 18, 2022, 23 pages, DOI: https://doi.org/10.3390/app10186527

[11] Dae-Neung Sohn, Jung-Tae Lee, and Hae-Chang Rim, “The Contribution of Stylistic Information to Content-Based Mobile Spam Filtering,” Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, 2009, 4 pages, https://aclanthology.org/P09-2081/

[12] Christophe Chong, Daniel Liu (Stanford), and Wonhong Lee (Neustar), “Malicious URL Detection”, Unspecified publication, 4 pages, https://cs229.stanford.edu/proj2012/ChongLiu MaliciousURLDetection.pdf

[13] Shiyun Li and Omar Dib, "Enhancing Online Security: A Novel Machine Learning Framework for Robust Detection of Known and Unknown Malicious URLs," Journal of Theoretical and Applied Electronic Commerce Research, vol. 19, no. 4, 2024, 42 pages, DOI: https://doi.org/10.3390/jtaer19040141

[14] Cho Do Xuan, Hoa Dinh Nguyen, Tisenko Victor Nikolaevich, "Malicious URL Detection Based on Machine Learning," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 1, 2020, 6 pages, DOI: http://dx.doi.org/10.14569/IJACSA.2020.0110119

[15] Salvi Siddhi Ravindra, Shah Juhi Sanjay, Shaikh Nausheenbanu Ahmed Gulzar, Khodke Pallavi, " Phishing Website Detection Based on URL," IJSRCSEIT, vol. 7, no. 3, 2021, 6 pages, DOI: https://doi.org/10.32628/CSEIT2173124

[16] Cheng Cao, James Caverlee, "Detecting Spam URLs in Social Media via Behavioral Analysis," Lecture Notes in Computer Science (LNCS), Springer, vol. 9022, 2015, 12 pages, DOI: http://doi.org/10.1007/978-3 319-16354-3_77

[17] Tasfia Tabassum, Md. Mahbubul Alam, Md. Sabbir Ejaz, Mohammad Kamrul Hasan, “A Review on Malicious URLs Detection Using Machine Learning Methods," Journal of Engineering Research and Reports, vol. 25, no. 12, 2023, 13 pages, DOI: http://doi.org/10.9734/JERR/2023/v25i121042

[18] Fuad A. Ghaleb, Mohammed Alsaedi, Faisal Saeed, Jawad Ahmad, Mohammed Alasli, "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning," Sensors, vol. 22, no. 9, 2022, 19 pages, DOI: https://doi.org/10.3390/s22093373

[19] Davide Canali, Marco Cova, Giovanni Vigna, Christopher Kruegel, "Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages," Proceedings of the 20th International Conference on World Wide Web (WWW 2011), 2011, 10 pages, DOI: https://doi.org/10.1145/1963405.1963436

[20] Daniel Schlette, Marco Caselli, and Gunther Pernul, “A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective,” IEEE Communications Surveys & Tutorials, 2021, DOI: http://doi.org/10.1109/COMST.2021.3117338

[21] Christophe Chong, Daniel Liu (Stanford), and Wonhong Lee (Neustar), “Malicious URL Detection”, Unspecified publication, 4 pages,

https://cs229.stanford.edu/proj2012/ChongLiu MaliciousURLDetection.pdf

[22] Muskaan V. Jaiswal and Anjali B. Raut, “Detecting and Blocking of Malicious URL,” International Journal of Science and Research (IJSR), vol. 10, no. 6, 2021, 3 pages, DOI: http://doi.org/10.21275/SR21610230148

[23] Malak Aljabri; Hanan S. Altamimi, Shahd A. Albelali, Maimunah Al Harbi, Haya T. Alhuraib, Najd K. Alotaibi, "Detecting Malicious URLs Detection Using Machine Learning Techniques: Review and Research Directions," IEEE Access, vol. 10, 2022, 23 pages, DOI: http://doi.org/10.1109/ACCESS.2022.3222307

[24] Nuria Reyes-Dorta, Pino Caballero-Gil, Carlos Rosa-Remedios, "Detection of Malicious URLs Using Machine Learning," Wireless Networks, vol. 30, 2024, 18 pages, DOI: https://doi.org/10.1007/s11276 024-03700-w

[25] Gunikhan Sonowal, "Detecting Phishing SMS Based on Multiple Correlation Algorithms," SN Computer Science, vol. 1, no. 361, 2020, 9 pages, DOI: https://doi.org/10.1007/s42979-020-00377-8

[26] The UCI SMS Spam Collection dataset https://raw.githubusercontent.com/justmarkham/pycon-2016-tutorial/master/data/sms.tsv

Author Profile

Aqsa Shaikh is a student of the PG program M.S. (Cyber Security) at the University Department of Information Technology, University of Mumbai. She is proficient in programming languages such as Java and Python, has knowledge in software development principles. The combination of her interest in cybersecurity and the groundwork provided by her hands-on experience with tools such as Wireshark and Nmap led her to seek a master’s degree specializing in cyber attacks and detection techniques. Her specialization includes Cryptography, Network Security, and Information Security. She has been introduced to security analysis, risk management, and vulnerability assessment concepts. One research paper related to cybersecurity has already been published by her. Now, her research work revolves around Smishing Detection, which is the practice of identifying and preventing SMS-based phishing attacks.

Mariya Shaikh, a postgraduate student, is pursuing her M.S. in Cybersecurity at the University Department of Information Technology, University of Mumbai, India. She has completed her bachelor's degree in Computer Science with a CGPA of 9.17 (A+). She has gained practical knowledge with internships in Cybersecurity, SOC operations, Digital Forensics, and OSINT. She has attended the training that is well-recognized by the industry from Microsoft, Cisco, and is currently holding ISC2 Candidate status. She has a TryHackMe ranking of Top 5%. Her deep interest in cybersecurity, together with practical experience with tools like Wireshark, Nmap, has led her to consider further studies in cyber attack detection and analysis. Her interests cover Network Security, Information Security, Digital Forensics, and OSINT. Her research interests are SMS phishing detection, machine learning-based security solutions, and cyber threat analysis.

Srivaramangai R., Head, Department of IT, University of Mumbai, India. Having 24 years of experience in teaching and 6 years in industry. The specialization areas are artificial intelligence, security, image processing. Has industry experience in web development and report code generators. Has published more than 35 International journal papers, 25 conference papers, resource persons for various workshops and chaired sessions. The papers relevant to Cyber Security includes “Assessment of Deep Packet Inspection System of Network traffic and Anomaly Detection”, Enhancing Security using ECC in Cloud Storage”, “Recapitulation of the Use of Machine Learning for Prevention of DDoS Attack on SDN Controller” and “Unmasking Deceptive Websites : Harnessing Machine Learning For Phishing Detection”.