IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DISTINGUISHING OF SPAM MESSAGES USING DEEP LEARNING WITH EMBEDDINGS

Sneha S
Department of CSE
JNNCE
Shivamogga,India

Dr. Jalesh Kumar
Department of CSE
JNNCE
Shivamogga,India

Abstract: Smishing, uses simple message service (SMS) for cyber-attacks, posing a significant threat on account of elevated likelihood of individuals providing personal information like account details and passwords, making it a critical issue. This project intends to build an effective framework for predicting spam or ham SMS messages using deep learning approaches and methods.A dataset of labeled messages is collected and pre-processed, and models like LSTM, Rnn, Bi-LSTM, and GRU are trained using GloVe word embedding techniques. The best-performing model is selected on performance metrics, optimized for robust performance, and deployed through a scalable pipeline.

Index Terms **- SMS, LSTM, Bi-LSTM,GRU and GloVe.**

I. INTRODUCTION

SMS has revolutionized communication but also introduces spam messages, posing security threats like phishing scams and malware distribution. Spam detection is crucial for maintaining digital communication systems' integrity and efficiency. Machine learning (ML) and Natural Language Processing (NLP) techniques, particularly Deep Learning (DL) models, effective in detecting SMS spam due to its limited character space and immediate delivery. A study developed robust, scalable models for spam detection and using embeddings, combining algorithms with deep feature extraction capabilities of neural networks . The projects aims to enhance trust in digital communication by ensuring users read non-spam messages cleanly and revamp the overall user experience by enhancing the effectiveness of a diverse dataset of labelled SMS and web messages.

II. LITERATURE SURVEY

Ghourabi, et al [1] have developed a hybrid CNN-LSTM model for SMS spam prediction in Arabic and English messages. The model, which combines CNN and LSTM deeplearning methods, outperformed traditional ML algorithms. However,it is not compatible with smartphones, URLs, files, messages, highlighting the need for improved detection methods for SMS spam.In [2] A hybrid DL model has been proposed to detect phishing techniques, which are common threats used by attackers to deceive users and obtain sensitive information. The model combines a CNN and LSTM to address this problem. Implemented on a benchmark dataset of 11430 URLs, the hybrid CNN with embedded features outperforms both models individually. Mishra, Sandhya, et al [3]. developed a neural network model for smishing detection, addressing a significant cybersecurity issue. The model achieved a final accuracy rate of 97.40%, with the URL feature being the most effective at 94%.Lee, Hwabin, et al [4]. presents a method using Visualization Technology and DL to predict spam messages across multiple languages.

The method combines text-processing and string-imaging techniques, using CNN 2D visualization technology to generate Unicode-based images. The image-based approach offers a comprehensive approach to spam detection, with an average accuracy of 0.9957. Shravasti et al., [5] introduce a groundbreaking a model called The "Smishing Detection: Using Artificial Intelligence" model uses advanced AI techniques to detect and mitigate smishing attempts in SMS, enhancing reliability and protecting users from potential data breaches. The LSTM-RNN DL model has a good accuracy rate, distinguishing between legitimate messages and smishing attempts. This approach is crucial for protecting users from potential cyber threats and improving efficiency by automating repetitive tasks. However, AI systems can also perpetuate biases in data. Mambina, Iddi S., et al., [6], developed a ML model to classify Swahili smishing attacks on mobile money users, aiming to improve security and reliability of mobile financial transactions in Sub-Saharan countries. The model uses preprocessed, tokenized, and word vectorized messages, with feature selection and parameter tuning applied using techniques like bag of words and n-gram. The hybrid model, combining ExtraTree classifier feature selection and Random Forest using TFIDF vectorization. However, it cannot be run on mobile devices. Balim, Caner, and Efnan Sora Gunal [7] This paper presents a machine learning (ML) system for detecting smishing attacks, a type of phishing that steals personal information from SMS messages. The system uses preprocessing, feature extraction, and classification stages to differentiate between legitimate and smishing messages. It offers a robust alternative to traditional methods, but lacks support for analysis in different languages, features, and classification algorithms, limiting its applicability in diverse regions. Alam, Mohammad Nazmul, et al [8] described Digital transformation has increased cybersecurity threats, particularly phishing attacks. Advanced ML techniques aim to improve detection and prevention, protecting users from financial and data losses. Anti- phishing software and dodging techniques are used, while threat intelligence and behavioral analytics detect unusual traffic patterns. An experiment used machine learning techniques to reduce data redundancy and identify variables' components. Mishra, Sandhya, and Devpriya Soni [9] proposes a Smishing Detector using SMS content analysis and URL behavior analysis to detect smishing attacks. The system aims to reduce false-positive results and enhance user information security. It comprises four modules: SMS Content Analyzer, URL Filter, Source Code Analyzer, and Apk Download Detector. The final prototype experiment showed a 96.29% accuracy, but the system is considered insecure for verifying application authenticity and may struggle with datasets with significant feature interactions. In [10],A CNN-LSTM-based prototype has been implemented for spam filtering mobile SMS, addressing the growing issue of SMS spam. Potentially enhancing smartphone security and reducing SMiShing attacks. However, the complexity of convolutional and LSTM layers makes it challenging to manage interactions and tune hyperparameters.

III. METHODOLOGY

In this section implementation of proposed SMS spam detection model, algorithm along with data exploration, processing and model generation has been discussed as shown below Fig.1.

Figure1:Data Flow diagram for proposed detection system

A. Exploring the data set: The Kaggle spam dataset is utilized for efficient classification and analysis due to its thorough exploration, understanding of data types, and ensuring data integrity.

B. Feature extraction: Feature extraction is crucial in data preprocessing, particularly in NLP and text analysis tasks. The Bag of Words model preserves word frequency but overlooks grammar and order.

C. Splitting the data into train and test: A train test split is a data analysis technique , where models are trained and tested using a specific number of training sets, with an epoch representing the series of sets used.

D. Model Generation : A prototype has been proposed to identify fraudulent messages using various algorithms, including MultinomialNB+TFIDF, multinomial NB+countvectorizer, simple rnn+Glove, GRU+Glove, and BI-LSTM+Glove.

1. **Multinomial Naive bayes with TF-IDF:** The method involves data collection, pre-processing, tokenization, normalization, and word removal. TF-IDF converts text data into numerical format, highlighting word importance. The algorithm calculates the likelihood of each word in a class and its prior probability. Bayes' theorem is used to compute posterior probabilities regarding prediction and the class with the highest probability is selected as output.
2. **Multinomial Naive bayes with CountVectorizer:** The approach involves data collection, pre-processing, tokenization, and CountVectorizer transformation. The text data is transformed into a sparse matrix of token counts, representing documents and unique words. By using a computation known as algorithm, it occur an assessment of the likelihood of each term in a category and the previous chances for all kinds. This process involves applying Bayes' theorem to calculate posterior probabilities for different categories then picking the one with highest value as predicted category.
3. **Simple RNN with GloVe:** The Simple RNN with GloVe approach involves data pre-processing, where text sequences are tokenized and mapped to numerical vectors using GloVe. GloVe provides dense, pre-trained word embeddings that enhance input data quality. A Simple RNN layer receives embeddings, which processes each sequence one token at the time while keeping an internal state for maintaining incoming data in order. To classify sequences, the final state that has been concealed is employed for extraction of features learnt. This mix uses GloVe's semantics plus RNN's learning power.
4. **LSTM with GloVe** : The LSTM with GloVe approach involves pre-processing text data using GloVe embeddings, which provide rich semantic information. This information is then fed into an LSTM network using gates (input, forget, and output) to manage information flow. The output often hidden state opinions themselves around the whole history therefore they are diverse in their nature as though it were a human being. These attributes make it perfect for such job positions including classification and sentiment analysis among others. In addition to this feature LSTMs have temporal anthropocentric features which makes it good in modeling language or even translating into another language apart from that which is being spoken at the moment.
5. **GRU with GloVe :** The GRU (Gated Recurrent Unit) with GloVe approach is a text data pre-processing method that uses GloVe embeddings to convert text into numerical vectors. A GRU network takes these embeddings as inputs, it applies an update and reset gates that control the direction of information and capture relationships over time series. After that, the output most often is passed through a dense layer for either classification or sequence prediction, wherein its common example is the last hidden state.This integration combines GloVe's semantic richness with GRU's efficiency in handling sequential data, making it effective for various NLP tasks.
6. **Bi-LSTM with GloVe** : The GRU (Gated Recurrent Unit) with GloVe approach is a text data pre-processing method that uses GloVe embeddings to convert text into numerical vectors. These embeddings are then input into a GRU network, which uses update and reset gates to manage information flow and capture dependencies in sequences. The output, typically the final hidden state, is then passed through a thick layer for tasks like classification or sequence prediction. This integration combines GloVe's semantic richness with GRU's efficiency in handling sequential data, making it effective for various NLP tasks.

IV. RESULTS AND ANALYSIS

The analysis of the models was done through a confusion matrix, ROC curve, accuracy, F1 score, recall, and precision. The F1 score is the weighted average of precision and recall while ROC curve represents trade-offs between sensitivity and specificityas shown in Fig.2.

	Model Name	Train Time	Test Time	Train Score	Test Score	Accuracy	F1	Precision	Recall	ROC-AUC
0	MultinomialNB + TFIDF	0.014188	0.000506	0.970738	0.958414	0.958414	0.825911	1.000000	0.703448	0.851724
1	MultinomialNB + CountVectorizer	0.005586	0.000571	0.989601	0.976789	0.976789	0.914894	0.941606	0.889655	0.940328
2	SimpleRNN + GloVe	8.184726	1.067338	0.990810	0.976789	0.976789	0.910448	0.991870	0.841379	0.920127
3	LSTM + GloVe	5.520883	0.377693	0.994438	0.977756	0.977756	0.919861	0.929577	0.910345	0.949548
4	GRU + GloVe	4.101050	0.161441	0.995405	0.972921	0.972921	0.904762	0.892617	0.917241	0.949622
5	Bi-LSTM + GloVe	4.032142	0.634958	0.993954	0.967118	0.967118	0.873134	0.951220	0.806897	0.900074

Figure 2: Performance Analysis of models

The potency of the models can be gleaned from bar graph performance analysis, identifying strengths and weaknesses, as illustrated in Fig.3, where scores and metrics are represented.

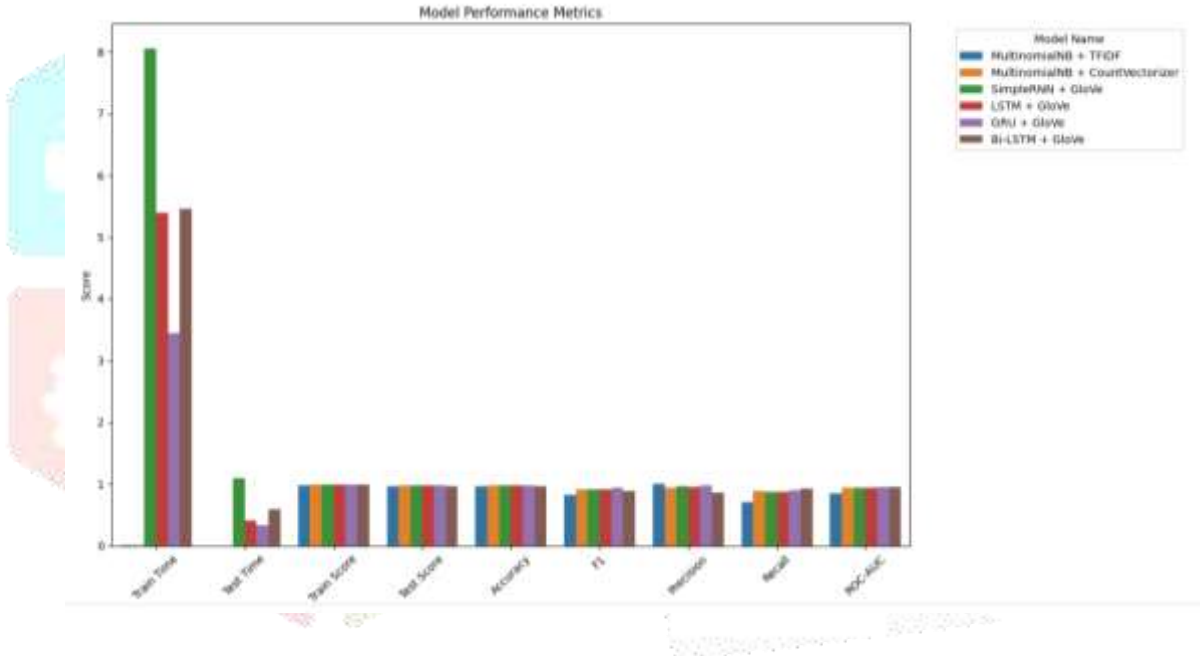

Figure 3: Grahical Representation of performance analysis of models

The GRU model and GloVe embeddings demonstrated exceptional performance in prediction tasks, achieving high accuracy, F1 score, recall, and precision, as illustrated in Fig.4.

```
inputs=["I will attend class tmr","You have won a scratch card please click here its urgent",
        "To use your credit,click the WAP link in the next txt message",
        "Tommorrow lets fix ice cream party"]
```

Output:

```
['Ham', 'Spam', 'Ham', 'Ham']
```

Figure 4 : Prediction Done using GRU+Glove

V. CONCLUSION

The project developed and tested a robust system for predicting spam or ham in SMS using ML and DL techniques, ensuring high-quality input data.The study analyzed various ML algorithms, DL models, and word embedding techniques, including Rnn, LSTM, Bi-LSTM, GRU, TF-IDF, Countvectorizer, and GloVe.The GRU+GloVe model was deemed the most effective in distinguishing spam from ham messages, achieving high performance metrics such as 97% accuracy, 89% precision, 91% recall, and 94% AUC-ROC.

REFERENCES

[1] Ghourabi, Abdallah, Mahmood A. Mahmood, and Qusay M. Alzubi. "A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages." *Future Internet* 12.9 (2020): 156.

[2] Zonyfar, Candra, Jung-Been Lee, and Jeong-Dong Kim. "HCNN-LSTM: Hybrid Convolutional Neural Network with Long Short-Term Memory Integrated for Legitimate Web Prediction." *Journal of Web Engineering* 22.5 (2023): 757-782.

[3] Mishra, Sandhya, and Devpriya Soni. "Implementation of 'smishing detector': an efficient model for smishing detection using neural network." *SN Computer Science* 3.3 (2022): 189.

[4] Lee, Hwabin, et al. "Visualization technology and deep-learning for multilingual spam message detection." *Electronics* 12.3 (2023): 582.

[5] Shravasti, Samyak Sadanand, and Manik Chavan. "Smishing detection: Using artificial intelligence." *Int. J. Res. Appl. Sci. Eng. Technol* 9.8 (2021): 2218-2224.

[6] Mambina, Iddi S., Jema D. Ndibwile, and Kisangiri F. Michael. "Classifying swahili smishing attacks for mobile money users: A machine-learning approach." *IEEE Access* 10 (2022): 83061-83074.

[7] Balim, Caner, and Efnan Sora Gunal. "Automatic detection of smishing attacks by machine learning methods." *2019 1st International Informatics and Software Engineering Conference (UBMYK)*. IEEE, 2019.

[8] Alam, Mohammad Nazmul, et al. "Phishing attacks detection using machine learning approach." *2020 third international conference on smart systems and inventive technology (ICSSIT)*. IEEE, 2020.

[9] Mishra, Sandhya, and Devpriya Soni. "Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis." *Future Generation Computer Systems* 108 (2020): 803-815.

[10] Hossain, Syed Md Minhaz, et al. "Spam filtering of mobile SMS using CNN–LSTM based deep learning model." *International Conference on Hybrid Intelligent Systems*. Cham: Springer International Publishing, 2021.

[11] D. Goel and A. K. Jain, ''Smishing-classifier: A novel framework for detection of Smishing attack in mobile environment,'' in Proc. Int. Conf. Gener. Comput. Technol., 2017, pp. 502–512.

[12] A. Aleroud, E. Abu-Shanab, A. Al-Aiad, and Y. Alshboul, ''An examination of susceptibility to spear phishing cyber attacks in non-English speaking communities,'' J. Inf. Secur. Appl., vol. 55, Dec. 2020, Art. no. 102614, doi: 10.1016/j.jisa.2020.102614.

[13] S. J. Delany, M. Buckley, and D. Greene, ''SMS spam filtering: Methods and data,'' Expert Syst. Appl., vol. 39, no. 10, pp. 9899–9908, Aug. 2012, doi: 10.1016/j.eswa.2012.02.053.