Spam Detection using NLP and Machine Learning Techniques

[1st] Sooraj Tomar
Department of Networking and Communications
SRM Institute of Science and Technology
Chennai, Tamil Nadu, India
*sooraj17tomar@gmail.com

[2nd] Harshit Krishna
Department of Networking and Communications
SRM Institute of Science and Technology
Chennai, Tamil Nadu, India
harshitkrishna2003@gmail.com

[3rd] Krishnaraj N
Department of Networking and Communications
SRM Institute of Science and Technology
Chennai, Tamil Nadu, India
krishnan2@srmist.edu.in

2023 International Conference on Emerging Research in Computational Science (ICERCS) | 979-8-3503-5976-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICERCS57948.2023.10433949

***Abstract*—Spam is one of the most common cyber-attacks in today's digital landscape. This paper compares spam detection capabilities of classical Machine Learning models using Inverse Document Frequency text processing with Deep Learning models using Corpus Indexing text processing using NLP and shows that Deep Learning models outperform classical ML models. As a result, SVM achieved 98.11% accuracy amongst the classical ML models whereas LSTM achieved an exceptional 98.85% accuracy. The BERT model achieved 99.29% accuracy, the highest for this use case. This paper discusses implementation, compares the various results from different authors and shows how Deep Learning helps to achieve the highest levels of accuracies possible for this use case.**

Keywords—Spam, Inverse Document Frequency, Corpus Indexing, SVM, LSTM, BERT.

I. Introduction

The term "cyber-attack" describes intentional and malevolent actions taken by people, organizations, or even nation-states with the goal of jeopardizing the security of computer systems, networks, devices, and data. To steal information, interfere with operations, obtain unauthorized access, or harm systems, these attacks take use of flaws and vulnerabilities in digital infrastructure. Cyber-attacks and the resulting damages have grown significantly in the past years as shown in Figure 1. [1]

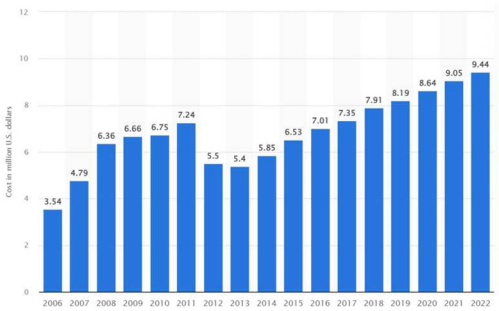

Fig. 1: Average cost of data breaches in the United States from years 2006 to 2022(in million USD)

According to an article on Britannica.com, E-mail has wrought one of the most common forms of cybercrime—spam, or unsolicited advertisements for products and services, which are estimated by experts to cover roughly 50 percent of all the e-mail circulating on the Internet. [2] Thus, Spam detection is a popular problem in today's cyber landscape. Spam detection has already been tackled in different ways by different authors to good levels of accuracy. The most popular way of preprocessing textual data is Inverse Document Frequency before sending this data to a Machine Learning Model. This paper uses the concept of Corpus Indexing to get one hot [10] representation of stem words of the textual data for preprocessing the data before feeding it into the Deep Learning Models. The Deep Learning Models are then trained for binary classification of spam messages from a given dataset. Finally, the results are compared from classical ML methods and Deep Learning methods using Inverse Document Frequency. This paper also discussed how this technique can also be used for detecting other text-based attacks like phishing and other social engineering attacks.

II. Literature Survey

Numerous academics have utilized machine learning and deep learning methodologies to identify spam messages from a provided dataset, specifically focusing on binary classification. The predominant techniques employed included TF-IDF (Term Frequency-Inverse Document Frequency) for data preprocessing, as well as models such as SVM and LSTM for classification. Kingshuk Debnath and his colleagues employed LSTM, BiLSTM, and BERT models, and obtained the best accuracy of 99.14% with BERT [3]. M.Rubin Julis et al. [4] employed various machine learning classifiers to attain a 98% accuracy rate using the Support Vector Machine model. Nilam Nur Amir Sjarif et al. [5] achieved an accuracy of 97.5% by applying the random forest classifier in combination with the term frequency-inverse document frequency data preprocessing (TF-IDF) technique. In their study, Pavas Navaney et al. [6] employed support vector machines and various other machine learning techniques to achieve an accuracy of 97.4%. Zainab Alshingiti et al. and Ishita Saha et al. both achieved high accuracy levels in detecting phishing and other text-related cyber-attacks using similar techniques. Zainab Alshingiti et al. achieved a 99.2% accuracy rate using CNN for phishing detection [7], while Ishita Saha et al. achieved a 95% accuracy rate using CNN for the same purpose [8]. However, none of them utilized Corpus Indexing methods, which can lead to better levels of accuracy.

III. Methodology

A. Dataset

The dataset used is an open-source dataset from Kaggle [9] and has been cleaned to get rid of any inconsistent entries and the columns have been named appropriately for better documentation. As is evident in Figure 2 and Table 1, the

dataset is imbalanced, just like a normal user's message inbox where more messages are of non-spam category than spam category. Category 0 denotes "Not Spam" and category denotes "Spam" message.

Table 1: Dataset Classification

Category	Count
0	4825
1	747
Total	5572

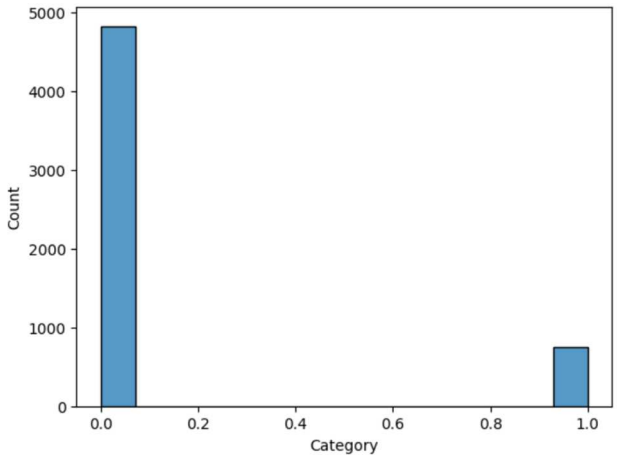

Fig. 2: Spam vs Not Spam rows in Dataset

B. *Data Preprocessing*

- *Term Frequency*: Inverse Document Frequency: The technique most commonly employed for such purposes and utilized by the other writers cited in this study. Token relevance within a document, compared to a collection of texts (usually a corpus), is assessed using a statistical measure. It takes into account two important factors: term frequency (TF), which quantifies how often a word appears in a document, and inverse document frequency (IDF), which evaluates how rare a word is in the total collection of documents. As the TF-IDF score increases, it indicates the increased significance of the term in describing the content of a document. Higher accuracies are achieved by classical ML models by applying this strategy.
- *Corpus Indexing:* This technique was applied in this paper for the Deep Learning models. It involves processes which systematically organize and catalogue textual data within a corpus, the corpus being a collection of relevant documents or texts which are to be analyzed. This method crucially enables efficient and effective information search and retrieval from large textual volumes of data. This method typically involves processes such as tokenization, where the text gets divided into singular words or tokens, and then these tokens get assigned specific identifiers such as in this paper, positions or indexes in an array. These identifiers then allow for rapid and precise retrieval of important data thus enhancing the speed and accuracy of various text analysis applications. Such techniques have evolved over time to incorporate advanced methods like stemming, lemmatization, and semantic indexing for even more sophisticated and context-aware information retrieval. Corpus indexing serves as a process to convert raw text data to the numerical input that deep learning models require, making it critical in various NLP and text analysis related applications. First, the words are broken down into their stem words before getting their one hot representation in terms of indexes from a custom corpus. These sequences of indexes are normalized by length and then given to the Deep Learning models for training and testing as shown by Figure 3.

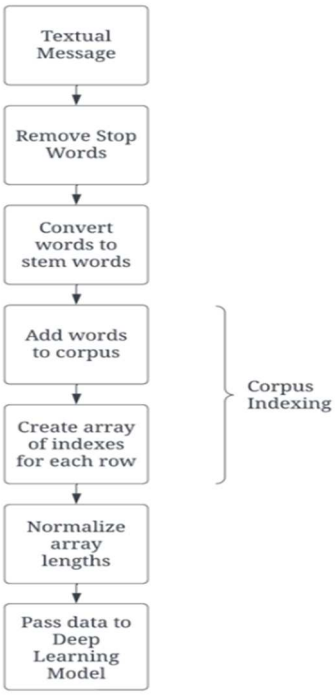

Fig. 3: Corpus Indexing Procedure

C. *Classcial ML Models Used*

This paper used TF-IDF scores for preprocessing data along with 5 classical ML Models for the binary classification of spam messages.

- *Support Vector Machine (SVM):* Support Vector Machines (SVM) are very effective in supervised machine learning applications utilized for solving regression and classification problems. To maximize the margin that denotes the separation between the hyperplane and closest data points from each class, this method determines the optimal hyperplane placement for classifying the data points into separate categories with high accuracy. A particular use of SVM is Support Vector Classification (SVC) which aims to identify a hyperplane that not only separates the given data into different categories but also maintains the maximum margin between these categories, making it exceedingly robust to outliers and adaptable to both linear and non-linear data.
- *Gaussian Naïve Bayes Algorithm:* GNB, a variation on the Naïve Bayes method, is a

popular probabilistic classical machine learning approach for classification tasks. It is best suited for scenarios where the feature variables are continuous and speculated to have a Normal (Gaussian) distribution. The assumption that the feature values for each class are normally distributed simplifies the probability calculations significantly but may lead to lower accuracy.

- *Light Gradient Boosting Machine (LGBM):* LGBM uses gradient boosting for high-performance. This framework was developed by Microsoft, specifically designed to efficiently handle voluminous datasets and high-dimensional feature spaces. The gradient boosting algorithm is employed, like other ensemble methods similar to XGBoost and AdaBoost, but uses a novel histogram-based approach for the splitting and growing of decision trees resulting in quicker training and reduced memory usage. This helps LGBM to achieve outstanding predictive accuracies and thus, LGBM is a powerful tool for several machine learning tasks, including but not limited to regression, classification, and ranking problems. It is handy in dealing with imbalanced datasets and categorical features without the need for extensive preprocessing.
- *Random Forest Classifier:* Random Forest Classifier is a powerful, adaptable and efficient machine learning method in both classification and regression applications. The reason behind is the combination of the predictions of several decision trees which functions as an ensemble learning technique. Random subsets of the training data and its characteristics are used to construct each one of decision trees. The unpredictability and generalization help the model to perform better; this also reduces overfitting. For a final categorization conclusion, a majority vote or weighted average of individual tree predictions is often used.
- *Decision Tree Classifier:* Decision Tree classifiers are versatile and intuitive and can be applied for both classification and regression tasks. By utilizing recursive division of the original dataset into subsets based on the input attribute values it classifies or predicts the target variable. The characteristic at each tree node is chosen to improve the purity or uniformity of the subsets with respect to the target variable.

D. Deep Learning Models Used

The textual data was preprocessed using Corpus Indexing before feeding it to the 3 used Deep Learning Models for binary classification.

- *Long Short-Term Memory (LSTM):* LSTM, a specialized type of recurrent neural network (RNN) which has proven itself to be remarkably effective in modeling and processing of sequential data, has the ability to capture long-range dependencies in text and can overcome the vanishing gradient problem thus, setting itself apart from the other models. LSTM achieves it through a sophisticated gating mechanism controlling the flow of information within its network's cells hence allowing the model to selectively store, update, or forget information over extended textual sequences. Thus, LSTMs are specifically preferred for a diverse range of natural language processing related tasks, time series analysis, speech recognition, and several other tasks. LSTM's ability to adapt to sequential data in combination with the capacity to maintain context over time, has established it as an essential element in the deep learning field. A base architecture of this model is shown in Figure 4.

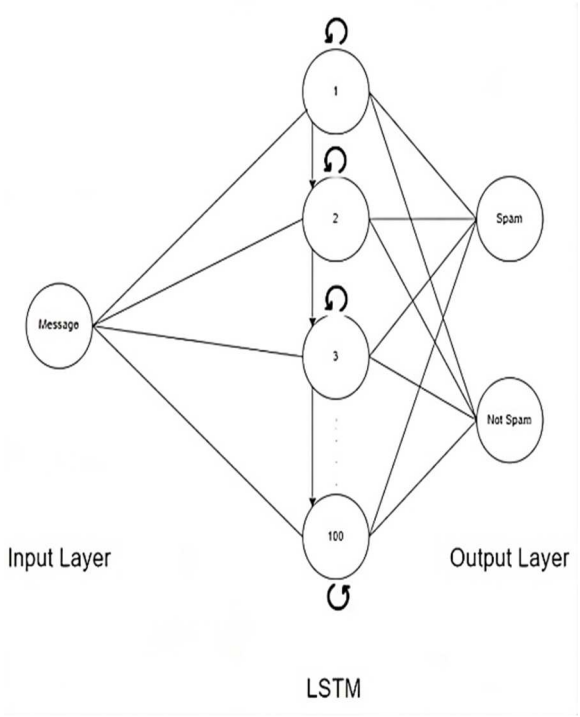

Fig. 4: LSTM Architecture

- *Bidirectional Long Short-Term Memory (BiLSTM):* BiLSTM is a recurrent neural network architecture which enhances LSTM's capabilities by analyzing sequences in both directions, forwards and backwards. BiLSTMs are excellent in capturing the context and interdependencies of sequential data by concurrently incorporating information from the past and future steps. By doing this bidirectionally the comprehending capacity of the model is improved and it can relate complex patterns, thus proving to be highly efficient in multiple natural language processing tasks such as sentiment analysis, named entity recognition, and machine translation. BiLSTMs enhance models' ability to acquire a more thorough comprehension of the context encompassing each piece in a sequence, hence enhancing their effectiveness on intricate tasks that involve sequences. BiLSTMs, like other deep learning models, necessitate a substantial volume of data in order to achieve effective generalization. When the dataset is of limited size, the model could tend to memorize the training samples instead of acquiring significant patterns, resulting in overfitting. A dataset that is not balanced can

result in decreased accuracy. Figure 5 illustrates a straightforward architecture of this paradigm.

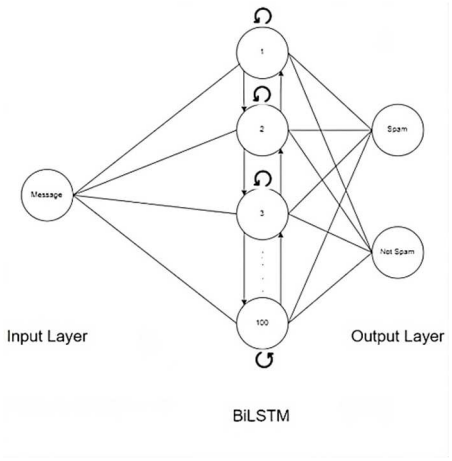

Fig. 5: BiLSTM Architecture

- LSTMs and BiLSTMs are quite similar in their base model implementations, as was the case in this paper as well shown in Figure 6.

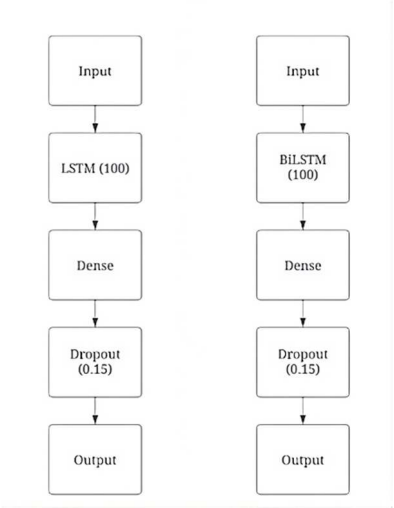

Fig. 6: LSTM and BiLSTM base models

- Bidirectional Encoder Representations from Transformers (BERT): BERT is remarkable in capturing contextual information from both the sides of a given word in a sentence, enabling it to grasp the full meaning of words in their context. This bidirectional context modeling has allowed BERT to achieve advanced levels of performance in a broad range of NLP tasks, including question answering, text classification, sentiment analysis, and language translation. BERT's pre-trained models, available in various sizes, can be tweaked for specific NLP tasks, making it a versatile and widely adopted tool for natural language understanding and generation. The model's large parameter size, sophisticated attention mechanisms, and transfer learning approach further contribute to its high accuracy, enabling it to generalize effectively to various NLP challenges. Overall, BERT's combination of extensive pretraining, contextual embeddings, and fine-tuning capabilities makes it a powerful tool for achieving revolutionary performance in language understanding and processing tasks. The only downside is that it is computationally expensive. A basic internal architecture of BERT is shown in Figure 7.

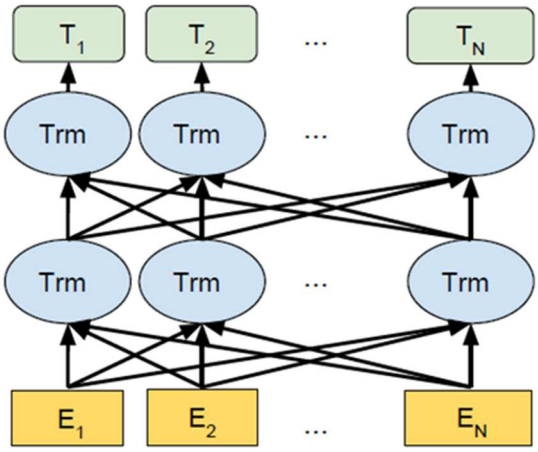

Fig. 7: BERT Architecture

IV. Results and Discussions

As of now classical ML models, LSTM and BiLSTM have been completely implemented whereas the trained BERT model is yet to be tested. There are different metrics which are used to judge the performance of the implemented models.

A. Metrics

- *Precision:* The accuracy of the positive predictions provided by a model. The calculation is derived from the ratio of True Positive to the sum of True Positive and False Positive.
- Recall, sometimes referred to as True Positive Rate (TPR), represents the proportion of accurately identified data samples belonging to a specific class out of the total samples for that class. The calculation is derived from the ratio of True Positive to the sum of True Positive and False Negative.
- The F1 Score is a metric that combines the precision and recall scores of a model to provide a measure of accuracy. The calculation is derived as follows: 2 multiplied by the product of Recall and Precision, divided by the sum of Recall and Precision.
- *Accuracy:* The proportion of accurate classifications accomplished by a trained machine learning model. The calculation is derived from the sum of true positive and true negative, divided by the sum of true positive, true negative, false positive, and false negative.

B. Classical ML Models

The textual data was preprocessed using Count Vectorizer Term Frequency – Inverse Document Frequency and then split into training and testing parts in an 80:20 ratio. The results formulated in Table 2 and Figure 8 show that SVM performed the best with an accuracy of 98.11% closely followed by LGBM classifier at 97.75%. The models were trained and

tested using Google Collaboratory with the general CPU Instance.

Table 2: Classical ML Models Results

Model	Accuracy (%)	Precision	F1 Score	Recall
Support Vector Machine (SVM)	98.11	0.93	0.96	0.99
Light Gradient Boosting Machine (LGBM)	97.75	0.93	0.95	0.97
Random Forest Classifier	97.58	0.91	0.94	0.99
Decision Tree Classifier	96.50	0.92	0.92	0.94
Naïve Bayes	89.14	0.89	0.81	0.77

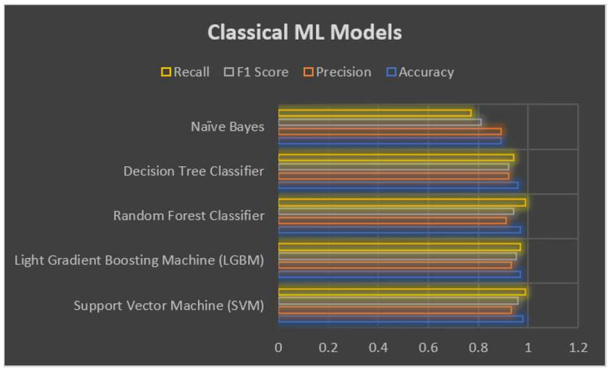

Fig. 8: Graph comparing performance of Classical ML Models

C. Deep Learning Models

The textual data was preprocessed first by removing the stop words, reducing the remaining words into their stem words and them preparing the corpus. The data was then turned into arrays of indexes using the corpus. This data was then split into training and testing parts in a 67:33 ratio. The results formulated in Table 3 and Figure 9 show that LSTM gave an exceptional accuracy of 98.85% followed by BiLSTM at 98.04%. BiLSTM gave a lower accuracy since it is complex and requires a better balanced and larger dataset. Hyperparameter tuning [11] and a dropout layer of 15% was used to mitigate overfitting of these models. The LSTM and BiLSTM models were trained and tested using Google Colaboratoty with the general CPU Instance. The BERT model gave a training accuracy of 99.76% and was trained using Google Colaboratoty with the T4 GPU Instance. Testing for the BERT model gave the highest accuracy at 99.29%.

Table 3: Deep Learning Models results

Model	Accuracy (%)		Precision	F1 Score	Recall
	Train	Test			
LSTM	98.04	98.85	0.99	0.97	0.96
BiLSTM	97.75	98.04	0.96	0.96	0.96
BERT	99.73	99.29	0.99	0.99	0.99

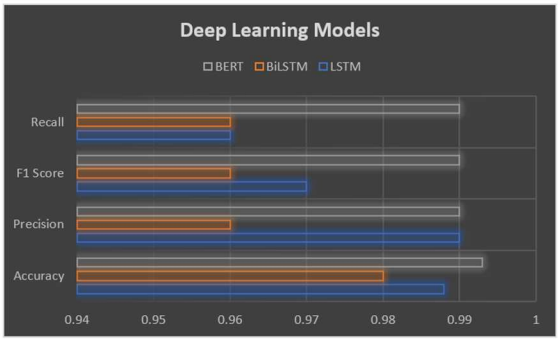

Fig. 9: Graph comparing performance of Deep Learning Models

A comparative analysis of the highest achieved accuracies for this use case by other authors to our results is shown in Table 4. Our LSTM model outperforms all other authors for this use case; however, the results of our BERT model are yet to be updated. As of now the highest accuracy has been achieved by Kingshuk Debnath et al. at 99.14% by their BERT model using TF IDF which is now surpassed this paper's BERT model at 99.29% using Corpus Indexing.[3]

Table 4: Comparison of Results of other authors and This Paper

Model	Accuracy (%)	
	Other Authors	This Paper
Classical ML	98.13 (MNB) [2]	98.11 (SVM)
LSTM	97.15 [2]	98.85
BiLSTM	98.34 [2]	98.04
BERT	99.14 [2]	99.29

This shows that Deep Learning models comfortably outperform classical ML models and preprocessing using corpus indexing is also a useful way of preprocessing data for Deep Learning models. The corpus can also be easily updated side by side the model for continued adaptability to new types of spam messages. This method can also be used to detect and filter out other types of textual content such as phishing emails, other social engineering attacks, fake news and even cyber bullying or derogatory comments.

V. Conclusion

The purpose of this paper was to apply corpus indexing for preprocessing of textual data and then feed that into Deep Learning models for highly accurate binary classification of a spam dataset. The results of the BERT model surpassed the highest accuracy achieved for this use case so far. A comparative analysis was also done in order to show that corpus indexing can also help get exceptional accuracies for

the same use case and it was also discussed that the same corpus indexing concept can help to solve other text related use cases as well. This work can be extended, with reasonable amount of effort, to classify spam messages in other languages also as the model will create its own corpus in order to understand the "context" of what features make a message spam or not spam.

REFERENCES

[1] Statistica "Average cost of a data breach in the United States from 2006 to 2023" July 2023.

[2] Britannica, The Editors of Encyclopaedia. "spam". Encyclopedia Britannica, 6 Oct. 2023, https://www.britannica.com/topic/spam. Accessed 16 October 2023.

[3] K. Debnath and N. Kar, "Email Spam Detection using Deep Learning Approach," 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 2022, pp. 37-41.

[4] Lakshmanarao, A. & Satyanarayana, S "SMS Spam Detection using Machine Learning and Deep Learning Techniques", Internatıonal Journal Of Scıentıfic & Technology Research, no. 02, 2020, pp 358-362.

[5] Sjarif, Nilam Nur Amir, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, and Suriani Mohd Sam, "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm." Procedia Computer Science 161, 2019, pp 509-515.

[6] Navaney, Pavas, Gaurav Dubey, and Ajay Rana. "SMS spam filtering using supervised machine learning algorithms," 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2018, pp. 43-48.

[7] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," Electronics, vol. 12, no. 1, 2023, p. 232.

[8] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana and S. Hossain, "Phishing Attacks Detection using Deep Learning Approach," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 1180-1185.

[9] Dataset Kaggle, https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[10] Hancock, J.T., Khoshgoftaar, T.M. Survey on categorical data for neural networks. J Big Data 7, 28 (2020).

[11] Lavesson, Niklas, and Paul Davidsson. "Quantifying the impact of learning algorithm parameter tuning." In AAAI, vol. 6, 2006, pp. 395-400.

[12] M. V. Koroteev, "BERT: a review of applications in natural language processing and understanding," in arXiv preprint arXiv:2103.11943, 2021.

[13] F.A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of BERT-based approaches," Artif Intell Rev, vol. 54, pp. 5789-5829, 2021.

[14] H. -T. Tseng, Y. -Z. Zheng and C. -C. Hsieh, "Sentiment Analysis using BERT, LSTM, and Cognitive Dictionary," 2022 IEEE International Conference on Consumer Electronics - Taiwan, Taipei, Taiwan, 2022, pp. 163-164.

[15] S. Kaddoura, G. Chandrasekaran, D. Elena Popescu, and J. H. Duraisamy, "A systematic literature review on spam content detection and classification," PeerJ Computer Science, vol. 8, pp. e830, 2022.