BERT Against Social Engineering Attack: Phishing Text Detection

Nafiz Rifat
Dept. of Computer Science
North Dakota State University
Fargo, ND, USA
nafiz.rifat@ndsu.edu

Mostofa Ahsan
Dept. of Computer Science
North Dakota State University
Fargo, NS, USA
mostofa.ahsan@ndsu.edu

Md. Minhaz Chowdhury
,
Dept. of Computer Science
East Stroudsburg University
East Stroudsburg, PA, USA
mchowdhur1@esu.edu

Rahul Gomes,
Dept. of Computer Science
University of Wisconsin-Eau Claire
Eau Claire, WI, USA
gomesr@uwec.edu

***Abstract*—Social engineering attack uses a wide range of human interaction tricks with the goal of achieving sensitive information. Certain tricks involve sending malicious SMS to the victim where the victim is convinced by the SMS, making a security mistake by clicking a malicious link or giving away confidential information. Machine learning algorithms for spam filtering are effective measures against such SMS spam. This paper demonstrated a novel universal spam detection model using pre-trained Google bidirectional encoder representations from Transformers (BERT) for classifying spam SMS in real-time scenarios. Subsequently, different classification techniques on the datasets are evaluated, based on their accuracy, precision, and recall. An overall accuracy reached 99%, with an F1 score of 0.97. The results and implications confirmed that the distilled BERT is an effective approach for spam detection.**

***Index Terms*—Phishing, SMS, BERT, Spam**

I. Introduction

SOCIAL engineering is a type of cyberattack where the attacker tricks the victim, with the goal of retrieving confidential information. The most common social engineering attack is phishing that uses counterfeit contents in emails or web sites to trick the victim [1]. A variation of phishing is SMS/text spam attack, also known as phishing texts or SMS spam attack [2]. Here, Short Message Service (SMS) is a procedure of sending short text messages from one device to another. An example of SMS spam phishing attack is, the victim receives a text message about phone bill and if they click the SMS given link, they can see the bill statement. When the link is clicked, the victim's mobile device is compromised, by downloading malicious software (jSMSHider malware installation, sending SMS to premium numbers by downloading Fakeplayer malware) [3], [4]. Another example is, the victim gets an SMS to provide her Personal identifiable information (PII), for a fake validation purpose. Attackers can also inject forged emergency messages e.g. Amber alert [5]. Mobile devices are vulnerable against such phishing attacks since users have a tendency to trust mobile devices, specially the SMS [6].

Many researchers have shown their effort in spam detection mechanisms, using available datasets to test spam detection algorithms. Different data-mining algorithms [7], [8] like Support Vector Machine (SVM) [9], Random Forest (RF) [10], [11], Associative Classification [12], intelligent models like Artificial Neural Networks [13], convolutional neural networks [14], and feature selection techniques [15] have proven effective. However, researchers are encountering an issue which is the scarcity of actual phishing website data compared to benign website data in training datasets. Therefore, there is a prospect to enhance the expected outcome. In this paper, we applied a combined approach to propose a solution to this issue. Apart from traditional machine learning, shallow machine learning algorithms are applied with a couple of BERT algorithms. Distilled BERT trained with neural network gives 98.63% accuracy for SPAM detection.

This paper is formatted in the following way: section 2 explains the related works, section 3 explains the dataset used in this paper, section 4 presents the data preprocessing process, section 5 describes the implemented machine learning algorithms, section 6 describes the results and outcome of the research and section 7 concludes the paper with suggested future research directions.

II. Related Work

Many researchers have put forth effort in developing spam detection systems. However, while reviewing other literature, we came across different approaches such as machine learning, deep learning, and combined techniques. In [8], after manually separating Ham and Spam and tokenizing SMS using tools to convert the text into an individual number of words and pre-processed data, the SVM classifier had more accurate results alongside the Naïve Bayes, random forest, and Decision tree techniques that produce high accuracy in their research without altering their original algorithm.

In [16], researchers used a procedure based on the frequency ratio, which calculates the lightness and quickness of filtering methods. It will allow filtering independently on the devices. They applied Naïve Bayes, Logistic Regression, and Decision Trees algorithms for their research and obtained a similar accuracy of around 94%. It proved that their proposed technique had a similar capability to others though it uses a simple calculation formula. In [17], [18] research was performed based on Naïve Bayes Classifier and Apriori Algorithm. However,

TABLE I: SMS messages data sample

Category	Message
ham	I wish that I was with you. Holding you tightly. Making you see how important you are. How much you mean tnaïvee ... How much I nnaïve you ... Naïvemy life ...naïv'
ham	Yup i'm still having coffnaïvewif my frens...'My fren drove shenaïve give me a lift...
spam	Sorry I'missed your call let's talk when'you have the time. I'm on 07090201529
ham	Unaïven say so early hor... U c already then say...
spam	URGENT! We are trying to contact U. Todays draw shows that you have won a å£800 prize GUARANTEED. Call 09050001808 from land line. Claim M95. Valid12hrs only

their performance relies on the statistical characteristics of the dataset.

Authors in [7] experimented with multiple classification algorithms. The datasets contained around 15% spam, and the rest are ham data amongst 5500 text messages. They processed the data with alphanumeric tokenization and generated 81,000 tokens from the short messages. Finally, out of thirteen classification algorithms, SVM achieved the best outcome.

While BERT with increased functionality obtained promising results to block fake COVID tweets [19], BERT encoding is becoming more widespread for extracting spam features, authors in [20] evaluated the performance with a minimal pre-processing and text cleanup. Eventually, the BERT model, coupled with various classification algorithms, achieved comparable to slightly better performances with BoW and TFIDF.

III. Datasets

This paper used a publicly available dataset - Spam Text Message Classification from Kaggle [21]. This dataset was developed by the same research group in [7]. The data contains structured data of SMS messages with the following columns category and message, showed in Table I. The message contains text messages and categories that contain if it is Spam or Ham. The dataset includes 5157 unique values, and amongst them, ham is 87% (4825), and spam is 13% (747). After the initial analysis, we get that the dataset is imbalanced; however, most available data are slightly imbalanced in the cyber security domain. We can use SMOTE [22], [23] to improve it, but it will generate synthetic data towards the upper class (Ham in our case), and we do not choose to have syntactic data in our dataset.

IV. Algorithms used

Selecting both method and parameter for any machine learning technique is essential to achieve high-performance levels from a predictive learning model [24]. In this paper, we are trying to optimize the complexity to deploy the model onto the Naive Bayes, Support Vector Machine, Logistic Regression, Word2vec, and Gradient Boosted Decision Trees (GBDT), compare the result with Bidirectional Encoder Representations Transformers (BERT).

A. Naïve Bayes

Naive Bayes classifiers are a group of classification algorithms established on Bayes' Theorem of conditional probability. It is a group of algorithms rather than a single algorithm that shares a common principle, i.e., every pair of features being classified is independent of each other. The model is simple to build and particularly useful for massive data sets. Naive Bayes is known to outperform even highly sophisticated classification methods with simplicity. Bayes networks are one of the most broadly used graphical models to represent and handle uncertain information [25].

Text classification techniques are commonly used for spam filtering [16]. As a result, there have been numerous studies on SMS text classification using Naïve Bayes Classifiers [26]. Using Bayes theorem, the probability of an event A occurring after an event B has already occurred, represented by $P(A|B)$ is calculated by using the formula:

$$\mathrm{P(A|B)} \;=\; \mathrm{(P(B|A)P(A))/(P(B))} \tag{1}$$

B. Support Vector Machines

Support Vector Machines (SVM) have obtained prominence in the field of machine learning and pattern classification, regression problems, and tasks like outlier detection. SVM was introduced by Vapnik and colleagues [9]. SVM has the potential to handle vast feature spaces because the training of SVM is carried out in a way so that the dimension of classified vectors enables detection of non-linear patterns.

Several instances of SVM are performing better than other classifiers by performance comparison [7]. However, they performed duplicate message analysis alongside SVM to achieve the goal. For a binary class problem, SVM typically performs a one-against-one calculation. A one-against-all technique is utilized for multi-label analysis [27]. However, this technique was performed by many researchers on multi-class problems [28].

C. Logistic Regression

Logistic regression is a popular regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, logistic regression is a predictive analysis. Logistic regression describes data and explains the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables.

This classification model is a linear classifier with a decision boundary of $\theta^{T} = 0$. Therefore, it is highly accurate when predicting probabilities rather than classes [29]. Some research [30] shows that in text classification, Logistic Regression is performed with high accuracy as the sample size is large with binary labels. However, on some occasions, Logistic Regression performed below par for text classification [31].

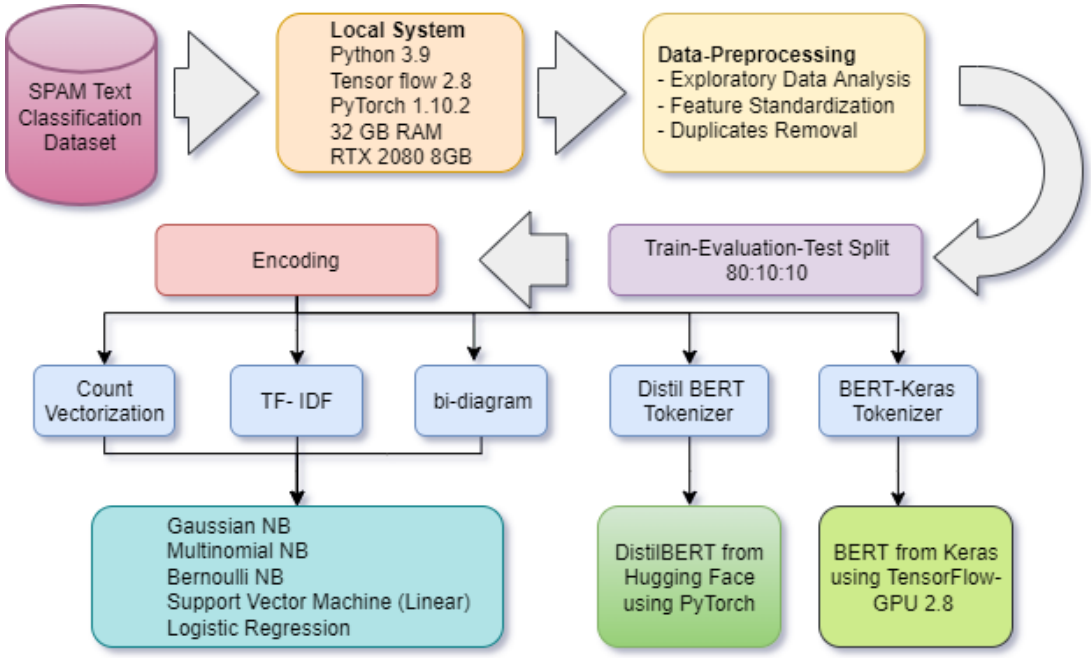

Fig. 1: Spam SMS/text Detection Process

D. Word2vec

Word2vec is a well-known sequence embedding approach that converts natural language into distributed vector representations [32]. It is frequently used as a preliminary step for predictive models in semantic and information retrieval tasks. It can capture contextual word-to-word interactions in a multidimensional environment.

The word2vec model has been widely applied in the segmentation of words [33], emotional classification [34], and POS Tagging [35]. The Word2vec model can train text vectors rapidly and efficiently. After training, the Word2vec model can map each text to a vector to represent their association. This sub-linear association into the vector space performed well in the research mentioned above.

E. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) was developed by Google and demonstrated state-of-the-art results on eleven NLP tasks [36]. BERT is a transformer-based NLP model designed to initially pre-train deep bidirectional representations from the unlabeled text by conditioning all layers on both left and right context. After the pre-training, calibration using labeled text can be done for other NLP tasks [36]. The BERT model was used to fine-tune text classification, and various experiments [36] carried out satisfactory results over the conventional approaches. The output is a vector representation that can be used to find similarities (distances) between spam and ham messages in the resulting embedding space. After applying conventional NLP approaches for our experiment, we apply BERT and compare the outcome amongst them.

V. Experiment

This section elaborates the details of the experimental setup and implementation of the research project to detect spam text. Figure 1 shows the process diagram of the implemented SMS/text detection process.

A. Environment Setup

We have created an environment with Tensorflow 2.8 on Python 3.9. The CudaToolkit used in this environment was 11.4.2. Our experiment also includes a PyTorch 1.10.2 virtual environment with Cuda Toolkit 11.3.0 for computation. Both of these environments utilize 32GB of DDR4 RAM and an RTX 2080 Max-Q 8GB GPU.

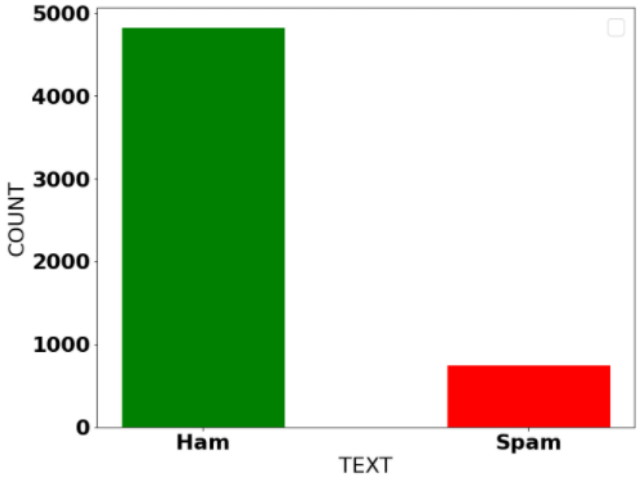

Fig. 2: Class Distribution

B. Exploratory Data Analysis

The Kaggle dataset only has one feature as text which contains Spam and Ham as labeled. The first criteria we noticed that the dataset id imbalanced class distribution as shown in Figure 2.

Only 13% of the whole data contained spam. But we did not perform any outlier removal techniques or down-sampling techniques on this dataset to simulate real-life cyber-attacks do not occur frequently. Hence, experiments on an imbalanced dataset were conducted to train the classifier to detect a threat in real-world scenarios. We noticed that the word density of texts those are labeled spam are unusually higher than ham texts, shown in Figure 3.

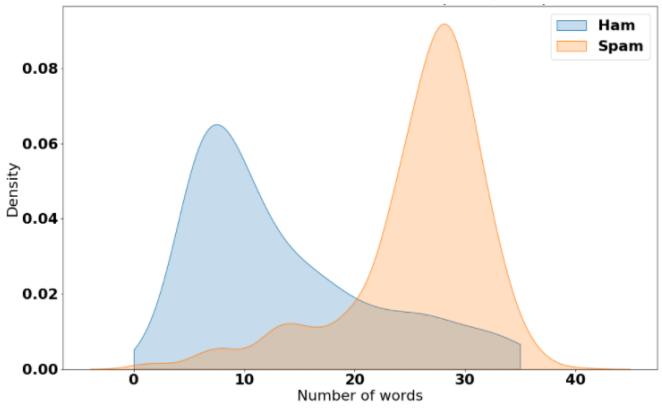

Fig. 3: World Density Comparison

This shows that the number of the word in ham messages are lower than the spam. Figure 4 shows us the message length of spam messages is also higher than that of hams. We discovered some interesting occurrences like the word "WIN" and "FREE" are very prevalent in spam messages which is clearly showed in Figure 5.

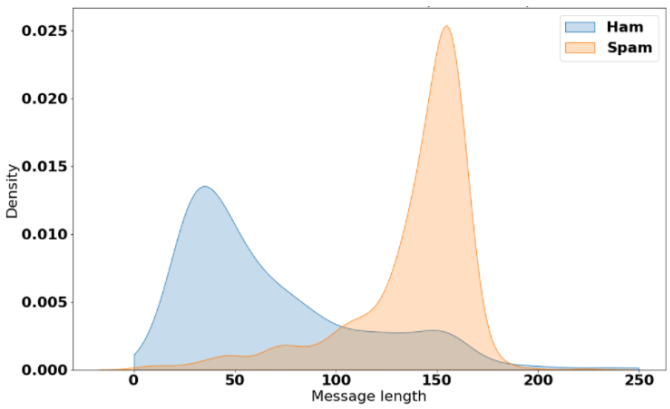

Fig. 4: Text Message Length

We analyzed the data to every character level and discovered that spam messages are usually clustered together. There is strong linearity for ham messages with a high number of uppercase characters, which is shown in Figure 6.

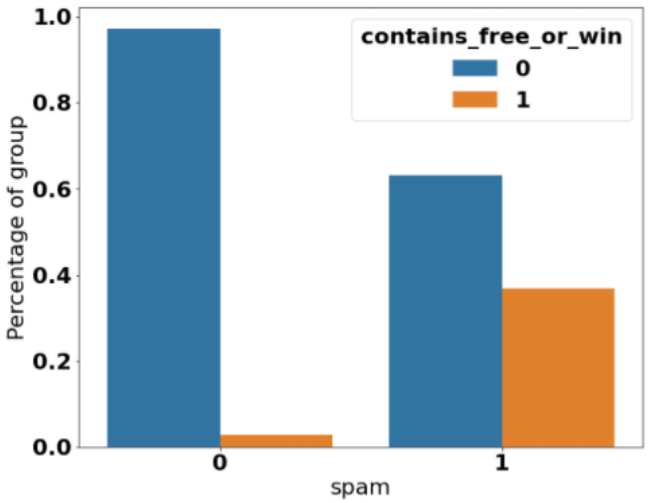

Fig. 5: Distribution Based on Word occurrences (free or win)

Before training the text messages directly using BERT, we encoded them using the DisitilBert tokenizer. After the encoding, we noticed that spam messages are generally closer together after embedding, shown in Figure 7.

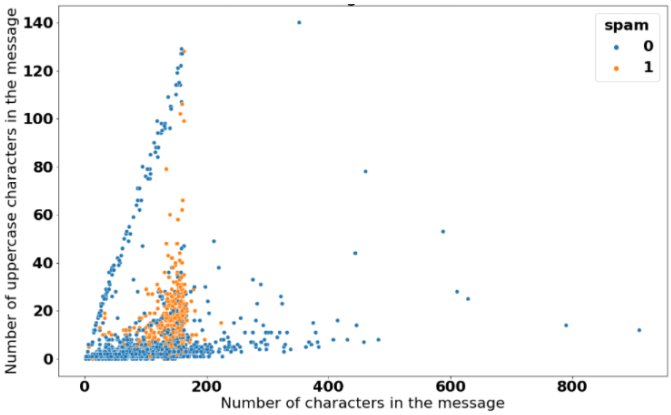

Fig. 6: Message Text Clustering Based on Character

C. Data Processing

The Kaggle Dataset had only one feature and one target column. The text column contains 5572 data points. 747 of these texts are labelled as spam and the rest 4457 texts are ham. At first step we labelled the spams as 1 and hams as 0. Data is then cleaned with multiple sequences of operations. We changed the words which have a pattern of email addresses to recognize them as similar followed by replacement of all the http links and web-address as singular website term. All the money symbols such as $, £ and USD etc were unified. Also, we have converted and replaced numbers as term "NUMBER". Finally punctuation was corrected and extra white spaces removed. After the data cleaning was completed, we created a full sample of the dataset in three different dataframe to perform Count Vectorization, TF-IDF matrix conversion and

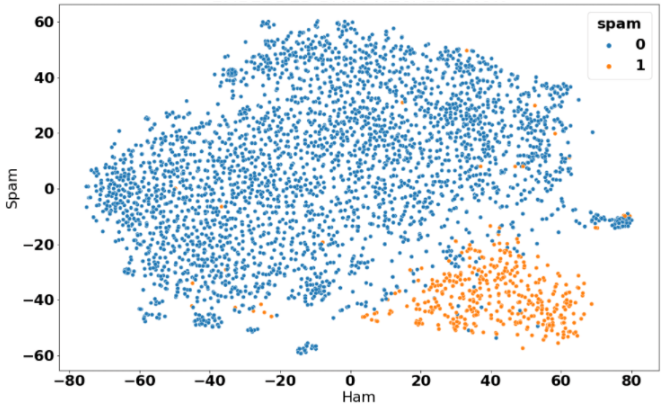

Fig. 7: Embedded Data Visualization

Bi-gram count vectorization. Each of these samples results in 100 maximum features based on english language stop words. Two types of tokenization were developed for different versions of the BERT classifier. The DistilBert Tokenizer is a pre-trained package that yielded the dataset to another 765 features, including the whole cleaned text. Later on, we added some new calculated features as Number of words, message length, number of upper case characters, number of uppercase words, etc. The final encoded dataset before training had 770 trainable features and one target column. We encoded the full sample of the dataset in a similar manner for BERT with Tensorflow as well, which yielded 769 trainable features after Tokenization using Bert-Keras package. For all of the experiments, dataset was split randomly as 80 percent for training, 10 percent for evaluation, and 10 percent for testing.

D. BERT Implementation

We have implemented both the Keras and DistilBert versions of BERT. We tokenized our dataset with different encoders according to their classifier package. Both of these classifiers used pre-trained models directly downloaded from Keras and HuggingFace. After encoding the dataset, we added extra neural network layers to train the model locally with pre-trained weights. A sequential model with 128X768 size was used as input for Bert_Keras classifier. All 769 features were used as direct input, and at the final layer, a dropout layer and sigmoid activation function was added. For the DistilBert experiment, we used the encoded data as 773 inputs and performed a couple of variations of multiple Dense layers with amounts of neurons. A neural network with the 1000-256-256-128-10-1 layers were used to train our encoded data. We used the RMSprop (Root Mean Square Propagation) as our optimizer and set the learning rate to 0.00001. The training was executed for 30 epochs with an early stop parameter enabled. RMSprop optimizer chooses different learning rate for each parameter. It also helps to automatically decrease the size of the gradient steps towards minima when there are large steps.

TABLE II: Performance Comparison of BERT variation

Evaluation Metrices	Distilled BERT with PyTorch	BERT with TensorFlow
Accuracy	0.99	0.92
Macro-Precision	0.97	0.92
Macro-Recall	0.97	0.92
Macro-F1 Score	0.97	0.92
Weighted-Precision	0.99	0.92
Weighted-Recall	0.99	0.92
Weighted-F1 Score	0.99	0.92

TABLE III: Performance Comparison of Shallow Machine Learning Algorithms

Algorithm Names	Count Vectorization Accuracy	RF-IDF Accuracy	Bi-gram Accuracy
Bernoulli NB	0.9389	0.9653	0.9389
Gaussian NB	0.9043	0.7715	0.9043
Multinomial NB	0.9389	0.9449	0.9389
SVM	0.9377	0.9659	0.9377
Logistic Regression	0.9264	0.6602	0.9264

VI. RESULT ANALYSIS

Many researchers have built spam classifiers using shallow machine learning models after encoding the texts with count vectorization, TF-IDF, and Bi-gram. The numerical inputs are easier to train by the shallow machine learning algorithms. Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic Regression have previously proven effective in detecting spam messages. Three encoding techniques were used to pre-process all the data before training them. Table II shows the performance comparison of BERT variation.

SVM with Linear kernel yielded the best accuracy using TF-IDF as 96.59%. When accuracy was compared with the proposed BERT implementation, the BERT-Keras yielded an accuracy of 91.87%. On the contrary, the DistilBert classifier yielded to 98.21%. The ROC-AUC resulted in 95.80%. The weighted average of the F1 score was 98%, and the macro average of the F1 score was 96%. These accuracy metrics are shown in Table III. The metrics prove that the DistilBert pre-trained model outperforms other models with limited cleaning and training on the text message to detect spam.

VII. CONCLUSION

In this paper, we applied shallow machine learning with a couple of BERT algorithms for SMS spam detection, that can be used against such social engineering (phishing) attacks. In addition, we summarized the recent improvements in SMS spam or phishing text filtering algorithms, mitigation, and detection approaches, as well as their limits and future research directions. Experiments were performed on imbalanced dataset to train the classifier. After implementing several encoding with different classification algorithms, results were documented and analyzed. Several algorithms achieved remarkable results like Multinomial NB with Count Vectorization achieved 93% accuracy, Bernoulli NB with RF-IDF was 96%, however, some low scores were also reported like Logistic Regression

with RF-IDF 62% accuracy. Finally, Distilled BERT trained with a neural network gives 98.63% accuracy for SPAM detection and outperformed all the other analyses; hence, it can be a good baseline for further comparison. To summarize, BERT algorithms have a great prospect in Spam and Intrusion Detection. Previously, it has received little attention in this type of research but is getting popular over time.

An important future work to consider is using different strategies to increase the dimensionality of the feature space. Data should be constantly added for precise analysis, as the SMS spam messages continuously increase over time. Our research can be taken to real-world application level to detect SMS spams and can prevent social engineering attacks.

REFERENCES

[1] Mike Mattera and Md Minhaz Chowdhury. Social engineering: The looming threat. In *2021 IEEE International Conference on Electro Information Technology (EIT)*, pages 056–061. IEEE, 2021.

[2] Protect against smishing, spam text messages, and text scams — verizon.

[3] Nikolay Atanassov and Md Minhaz Chowdhury. Mobile device threat: Malware. In *2021 IEEE International Conference on Electro Information Technology (EIT)*, pages 007–013. IEEE, 2021.

[4] Aaron Mos and Md Minhaz Chowdhury. Mobile security: A look into android. In *2020 IEEE International Conference on Electro Information Technology (EIT)*, pages 638–642. IEEE, 2020.

[5] John A Khan and Md Minhaz Chowdhury. Security analysis of 5g network. In *2021 IEEE International Conference on Electro Information Technology (EIT)*, pages 001–006. IEEE, 2021.

[6] Mariantonietta La Polla, Fabio Martinelli, and Daniele Sgandurra. A survey on security for mobile devices. *IEEE communications surveys & tutorials*, 15(1):446–471, 2012.

[7] Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262, 2011.

[8] Hassan Najadat, Nawaf Abdulla, Raddad Abooraig, and Shehabeddin Nawasrah. Mobile sms spam filtering based on mixing classifiers. *International Journal of Advanced Computing Research*, 1:1–7, 2014.

[9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[10] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[11] Rahul Gomes, Mostofa Ahsan, and Anne Denton. Random forest classifier in sdn framework for user-based indoor localization. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0537–0542. IEEE, 2018.

[12] Fadi Thabtah. A review of associative classification mining. *The Knowledge Engineering Review*, 22(1):37–65, 2007.

[13] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[14] Rahul Gomes, Papia Rozario, and Nishan Adhikari. Deep learning optimization in remote sensing image segmentation using dilated convolutions and shufflenet. In *2021 IEEE International Conference on Electro Information Technology (EIT)*, pages 244–249. IEEE, 2021.

[15] Mostofa Ahsan, Rahul Gomes, Md Chowdhury, Kendall E Nygard, et al. Enhancing machine learning prediction in cybersecurity using dynamic feature selector. *Journal of Cybersecurity and Privacy*, 1(1):199–218, 2021.

[16] Ye-wang CHEN and Jin-shan YU. An improved text classification method based on bayes. *Journal of Huaqiao University (Natural Science)*, 2011.

[17] S Sable and PN Kalavadekar. Sms classification based on naïve bayes classifier and semi-supervised learning. *International Journal of Innovations in Engineering Research and Technology*, 3(7), 2016.

[18] Ishtiaq Ahmed, Donghai Guan, and Tae Choong Chung. Sms classification based on naive bayes classifier and apriori algorithm frequent itemset. *International Journal of machine Learning and computing*, 4(2):183, 2014.

[19] Debanjana Kar, Mohit Bhardwaj, Suranjana Samanta, and Amar Prakash Azad. No rumours please! a multi-indic-lingual approach for covid fake-tweet detection. In *2021 Grace Hopper Celebration India (GHCI)*, pages 1–5. IEEE, 2020.

[20] Sergio Rojas-Galeano. Using bert encoding to tackle the mad-lib attack in sms spam detection. *arXiv preprint arXiv:2107.06400*, 2021.

[21] Sms spam collection dataset — kaggle.

[22] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[23] Mostofa Ahsan, Rahul Gomes, and Anne Denton. Smote implementation on phishing data to enhance cybersecurity. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0531–0536. IEEE, 2018.

[24] Sarah Jane Delany, Mark Buckley, and Derek Greene. Sms spam filtering: Methods and data. *Expert Systems with Applications*, 39(10):9899–9908, 2012.

[25] Finn V Jensen et al. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[26] Tej Bahadur Shahi, Abhimanu Yadav, et al. Mobile sms spam filtering for nepali text using naïve bayesian and support vector machine. *International Journal of Intelligence Science*, 4(01):24–28, 2014.

[27] Rami M Mohammad, Fadi Thabtah, and Lee McCluskey. An assessment of features related to phishing websites using an automated technique. In *2012 International Conference for Internet Technology and Secured Transactions*, pages 492–497. IEEE, 2012.

[28] Gou Bo and Huang Xianwu. Svm multi-class classification. *Journal of Data Acquisition & Processing*, 21(3):334–339, 2006.

[29] Alexander Genkin, David D Lewis, and David Madigan. Large-scale bayesian logistic regression for text categorization. *technometrics*, 49(3):291–304, 2007.

[30] Waqas Haider Bangyal, Rukhma Qasim, Zeeshan Ahmad, Hafsa Dar, Laiqa Rukhsar, Zahra Aman, Jamil Ahmad, et al. Detection of fake news text classification on covid-19 using deep learning approaches. *Computational and Mathematical Methods in Medicine*, 2021, 2021.

[31] Tomas Pranckevičius and Virginijus Marcinkevičius. Application of logistic regression with part-of-the-speech tagging for multi-class text classification. In *2016 IEEE 4th workshop on advances in information, electronic and electrical engineering (AIEEE)*, pages 1–5. IEEE, 2016.

[32] Haixia Liu. Sentiment analysis of citations using word2vec. *arXiv preprint arXiv:1704.00177*, 2017.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[34] Bai Xue, Chen Fu, and Zhan Shaobin. A study on sentiment computing and classification of sina weibo with word2vec. In *2014 IEEE International Congress on Big Data*, pages 358–363. IEEE, 2014.

[35] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 647–657, 2013.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.