Detection of Phishing in Mobile Instant Messaging using Natural Language Processing and Machine Learning

Academic Internship
MSc Cybersecurity

Suman Verma
Student ID: x21154996

School of Computing
National College of Ireland

Supervisor: Dr. Vanessa Ayala Rivera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Suman Verma

Student ID: x21154996

Programme: MSc Cybersecurity **Year:** 2022

Module: Academic Internship

Supervisor: Dr. Vanessa Ayala Rivera

Submission Due Date: 15/12/2022

Project Title: **Detection of Phishing in Mobile Instant Messaging using Natural Language Processing and Machine Learning**

Word Count: 7654 **Page Count:** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Suman verma

Date: 28/01/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Phishing in Mobile Instant Messaging using Natural Language Processing and Machine Learning

Suman Verma
x21154996

Abstract

Advancement in mobile technology has made communication possible in real time with much ease but at the cost of wider attack area available for phishing. Detection of phishing in instant message is a matter of concern and research due to its widespread use for personal, professional, and business purpose. Cyber attackers are gradually modifying the modus operandi of phishing since its inception from worms, virus, malicious link to wise use of languages invoking fear, urgency, reward, in instant messages for mobile users. There has been continuous research being done to detect phishing in E-mail and SMS using advance technologies but detection of phishing in Instant message remains neglected. The widespread usage of instant messengers by individuals of all ages, including the most susceptible groups like the elderly and younger generations, necessitates the addition of security features for phishing detection and message filtering. This research is aimed at detecting phishing in mobile instant messages by analysing the language of message with the help of Natural Language Processing and building a classifier to detect the keywords pointing towards phishing. Indication of phishing messages cannot be limited to direct use of question or command to users as the language of message can be modelled, depending on the context and emotional state of users during real-time conversation. The SMS Phishing dataset from Mendeley data dedicated for machine learning and pattern recognition was employed in our research since the keywords used in the dataset and the machine learning technique were pertinent to our study. The dataset has been pre-processed before training the classifier. To compare the better vectorisation methods for feature extraction, three different techniques namely Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TFIDF) and Word2vec has been applied on the pre-processed data. Three classification models-Random Forest, Logical Regression and Gaussian Naïve Bayes are trained on the dataset for identification and classification of messages into phishing and legitimate messages. Our tests showed that using TFIDF for vectorization and trying to balance the data with Random over sampling increased classifier performance. Random Forest classifier predicted the messages into phishing and no phishing with accuracy of 99.2 % among three models on the dataset. With a dataset devoted to instant messages, the Word2vec method of vectorization might further increase its classification accuracy, which was 95.2% when trained on Random Forest classifiers. It is necessary to create a dataset for instant messaging that would show contextual relationships between sentences, variations in linguistic structure utilized for phishing, or pretexting for phishing to detect it. Proactive detection of phishing in instant messages will have a pivotal role for a large fraction of society and organisation to safeguard the application as well as valuable customer.

Keywords- Instant Message, Social Engineering, phishing detection, Natural Language Processing, BOW, TFIDF, *Word2vec, Logical Regression, SVM, Random Forest, Gaussian Naïve Bayes*

1 Introduction

Phishing is a crime that involves tricking people into giving up their personal and financial information through social engineering and technology (APWG 2022). Due to the development of applications for all purposes, mobile phone usage has increased dramatically with technological advancement. Recent decade has seen increased use of Mobile Instant messaging applications like WhatsApp, Telegram, and the same way these have been target of social engineers for phishing scam but there has been lack of interest to study phishing susceptibility in these applications. Ahmad, R., and Terzis (2022) recommended that users of these instant messaging applications click on links without hesitation and forward them to friends and groups, which could result in many individuals falling for it because it comes from their known contacts.

Considering the report of Anti-Phishing Working Group, the second quarter of 2022 saw 1,097,811 phishing assaults in total, setting a record and ranking as the worst quarter for phishing and even Smishing and vishing incidents rose in Q2 2022, indicating a rise in mobile-based fraud (APWG 2022). Most common vector of phishing is the malicious e-mail sent by an impersonator, from employer, service provider or randomly targeted, all aiming to extract crucial information such as credentials or account details. Phishing aimed at instant message is more potent as it could target anyone from the society even the vulnerable group of society, due to the availability of mobile phones to everyone including children especially after corona epidemic in recent times. Phishers innovate new methods and with text messages, its easy to manipulate the intention in words to befriend first, gain trust and then apply the phishing scam.

At present, most of the available tools and research area to detect phishing is more concentrated towards E-mail, Spam SMS identification, while detection of the social engineering through conversation in messages remains neglected (Salahdine & Kaabouch, 2019). In some instances, small screen of mobile makes it difficult to observe all the necessary feature in a short span of time and thus victim falls prey to the lure of attacker (Jain et al., 2022). Instant message conversation can be in real time and can be very much influential, personalised towards victim in sense of psychological state of mind (A. Stone, 2007). Therefor while finding solution for phishing attack through instant message, the analysis of text message plays a crucial role. Human language is complex and can have varied interpretation depending upon the context of situation while in conversation. Due to this, we are presenting an approach through this research proposal using machine learning and natural language processing to identify phishing messages in in instant messaging and label them as such.

Research Question: The above research problem motivates the following research question: "How to improve effectiveness of phishing identification in instant messaging using NLP based on machine learning?"

The goal of this study is to analyse mobile instant messaging for terms that could entice users into disclosing private information, clicking on harmful links, downloading executable files, or engaging in prohibited behaviour that could result in phishing attacks. The SMS Phishing dataset from Mendeley data has been subject to the following approach for this research purposes utilizing NLP and ML models. Three techniques of vectorization have been applied to extract the features for training the classifiers. To train the supervised models, the vectorized dataset has been divided into train and test sets. The model accuracy has then been tested using the test set of the dataset. Logical Regression Classifier, Gaussian Naive Bayes Classifier, and Random Forest Classifier Natural have been experimented with three techniques of vector representation namely Bag of words, TFIDF and Word2vec. The results have been evaluated by comparing and analysing the accuracy of classification and recall percentage of phishing messages as presented by ML models using different vectorisation methods. The classifiers performed well with accuracy over 84% when trained and tested with document matrix using Bag of words. When the dataset was balanced with ROS and vectorized using TFIDF, Random Forest Classifier outperformed other models with accuracy for phishing detection of 99.2% and recall of 85%. Word2vec method of word embedding when used with Random Forest Model performed the classification with an accuracy of 95.2% with the dataset.

Integration of NLP and Machine Learning Classifier as a model in the mobile instant messenger application promises the detection and blocking of phishing messages for the benefit of message provider and public in general.

The research outlines and discuss the work done to detect phishing in Mobile Instant messages in section 2 of Literature review. The research methodology has been discussed in section 3. Section 4 describes the design specification of future work to develop a model to detect phishing when integrated with the current research done. The implementation of current research work performed has been detailed in section 5, whereas section 6 presents the evaluation and discussion of the research performed. Section 7 details the comparative result of the research conducted, and Section 8 concludes the research with pointers discussed for future work in the area.

2 Literature Review

Detection of phishing in communication media, if left solely on the shoulders of human could be flawed because of fragile nature and other factors that consider human being as weak link and susceptible to social engineering attacks. This may not be true, but this problem could be fixed by training the system(software) to detect phishing rather than the end users. Thus, inclusion of phishing detection model by the instant message provider would be great idea to eliminate the risk of phishing.

Phishers attempt to act as reliable conversation partners on instant messengers and request personal and sensitive information to get access to the unaware victim's sensitive information, such as passwords and security codes for bank accounts. Rana Alabdan (2020) pointed that instant messaging contained emojis, images, gifs, files, hyperlinks, and more, in addition to text. Further some of the instant message application supported voice and video calls, thus the ideal environment for social engineers. As the name implies, instant messaging enabled real-time communication, enabling phishers to engage with targets and coerce them into disclosing personal information through various scams. Given that criminal tactics and methods of operation are well known, as well as the fact that phishing incidents are routinely made public, the issue of what has been done to stop this nuisance naturally emerges. Without a doubt, research is being done to determine how to counter this threat, and new technologies are being developed to strengthen our defences while every available option is being tested against it.

2.1 Training to end users

Various studies have assessed the efficacy of updating the system and users as one method to stop phishing at any level. It is usually advisable to offer training to end users and users at all levels when it comes to upgrading users. J. Scheeres, (2008) suggested training for people that might provide them real-world experience with social engineering tactics than just warning about the dangers, workable solutions, and recommendations. The training must consider and eliminate the user's individual vulnerability for it to be effective. Boateng & Amanor (2014) pointed out that only 35% users they researched were aware of threats against mobile devices such as phishing, vishing and smishing and thus could be easy targets. They further remarked in their study that men were more susceptible to attack than women on the facts and figures of men being more technical savvy and thus found themselves comfortable in establishing trust in cyber space easily. They proposed to advertise dictionary of probable "intriguing" and "deceptive" terms used in phishing attempts for end users as a starting point to avoid falling for these social engineering attacks.

Overall, these studies towards training to end users regarding phishing attacks highlight the need for a well-structured and practical approach that will not only highlight the damage but at the same time equip them with the knowledge to identify and report it.

2.2 Content analysis and URL behaviour

Guan et al. (2009) proposed an anomaly-based approach of URL and user behaviour to detect phishing in instant messages which could contain a malware file or link to malicious website. They approached the proposal by matching the URL with known malicious behaviour and developed score model to predict the malicious URL in real time. C. Singh and Meenu (2020) proposed to detect phishing in e-mail and messages with the presence of malicious link in it. The research applied machine learning to detect malicious URL features like long length of URL, shortened URL, presence of @, extra dots etc. Mishra and Soni (2020) too presented an approach to detect smishing in SMS through content analysis and URL behaviour termed as ‘Smishing Detector’. The proposed model consisted of four modules as SMS analyser, URL filter, Source code analyser and Apk downloader. NLP and Machine learning models were used to detect malicious intent and keyword in content of messages that contained an e-mail or phone number. The result showed accuracy of 96.29% using Naïve Bayes classifier and model provided far better security than others.

The above works does reinforce the idea of content analysis of e-mail, SMS, and instant messages only on presence of malicious link which could not true and helpful in current scenario.

2.3 Data Mining approach to detection of phishing

Ali and Rajamani (2011) proposed detection of phishing in instant message using data mining technique of association rule mining and information retrieval technique. They demonstrated an association rule mining method (Apriori algorithm) to identify fraudulent phishing and suspicious communications delivered through instant messaging between multiple clients. The instant messages targeted used for the research had phishing attempts, where attackers attempt to learn the password and other security-related information through queries. Quaseem S. M. and Govardhan. A. (2014) refined the above-mentioned work for phishing identification in instant messages using Classification Based Association (CBA) and Domain Ontology. They further stated that phishing identification could be ascertained by determining the context of the phishing terms identified. These when combined with retrieved domain can produce more intriguing phishing rules that will produce instant, dynamic phishing alerts for the victim with the highest possible performance, or maximum true positives percentage based on Classification Based Association (CBA) rules.

Together these studies have researched to detect phishing in instant messaging application and presented promising result.

2.4 Natural language Processing and Machine Learning to detect phishing

Sawa et al. (2016) suggested using natural language processing to analyse conversation in instant messages to extract any query or command to extract these as likely themes to detect social engineering. To determine if they were malicious or not, they were also compared with topic backlists. In conjunction with Sawa et al., (Peng et al. 2018) also suggested the same strategy as previously stated, with semantic analysis of the text of instant messages using natural language processing. Working with a Spanish dataset Lopez, J.C., and Camargo, J.E.,

(2022) developed this strategy further by extracting more characteristics from the text for analysis with an accuracy of over 80% utilizing Neural Networks, SVM, and Random Forests. The research focused on one-way and two-way communication, and the user reports that contain plain text are utilized to compile the data on social engineering attacks from various sources such as e-mail, SMS, chats.

Yuanyuan Lan (2011) proposed a model to detect social engineering attack during conversation using NLP and Deep learning. It suggests a social engineering assault detection model that processes dialog context using attention-based Bi-LSTM and fuses user characteristics and text characteristics together using ResNet model. The viability and effectiveness of the proposed model are shown by examining the approach objects' properties in terms of the rationale and application of algorithm selection.

Bountakas et. al. (2021) performed research to compare several NLP techniques namely (TF-IDF, Word2Vec, and BERT) and ML approach combinations for the detection of phishing emails to determine the best combination based on performance. The findings of research indicated that the Word2Vec methodology performed best with ML phishing email detection strategy on text content of the mail. Particularly, the Word2Vec along with Random Forest and Word2Vec along with Logical Regression combinations produced the best outcomes in the balanced and imbalanced datasets, respectively.

A. Kovač, et.al. (2022) presented review of various machine learning algorithms for detection of phishing in electronic messaging service with average accuracy over 90%. Support Vector Machine (SVM) and Random Forest outperformed others with 99.87& accuracy. The dataset size, source and train-test size varied across the research. Nitish Sharma (2022) presented the study on different machine learning models using different of pre-processing steps on dataset and TF-IDF for feature extraction. Random Forest and Gradient boost outperformed other models with all pre-processing steps. It advised use of Word2vec as feature extraction for future research.

Together, these investigations exploring the application of NLP in phishing detection primarily focused on the text contents of emails or SMS messages and showed promising results. This encourages us to use NLP and ML as the baseline criteria for phishing detection on mobile instant messages.

3 Research Methodology

The proposed research methodology was to build a classification model to detect phishing in mobile instant messaging application using the natural language processing and machine learning. The mobile instant messaging could be used by social engineers as phishing vector, leading to disclosure of confidential and restricted information either directly or indirectly. Phishing could be attempted in various known ways or innovating, and modelling language used in messages to compel the victim to disclose important information. In addition to this, messages could have blacklist mobile numbers, spoofed hyperlinks or backlist e-mail prompted for further communication. The confidential information if provided could be used for phishing attack. Detection of phishing through direct command, question, malicious link, blacklist phone number and e-mail were thoroughly researched but the indirect method of phishing using language variation had been quite neglected mainly in mobile instant messaging. This provided the motivation to build a classification model for detection and isolation of phishing messages by analysing instant messages using NLP and machine learning models. For the above mentioned purposed, we analysed the “SMS phishing dataset” and the research methodology included the following steps, which will be described in more detail in the next paragraphs:

- **Dataset**- Open-source dataset of SMS phishing from Mendeley data (Mishra and Soni) had been used for the research.
- **Cleaning and Data pre-processing**-Dataset to be used had been cleaned and pre-processed to be utilized for machine learning model.
- **Vectorization**-Representation of processed text into machine readable form had been performed by three different methods of vectorization namely-Bag of words, TFIDF and word2vce.
- **Building the classification model**- Three supervised machine learning classifiers had been trained and evaluated on the dataset were Random Forest, Logical Regression and Gaussian Naïve Bayes.
- **Cross validation and Evaluation**- The different classifiers had been cross validated, and parameter tuned for evaluation and selection of model.

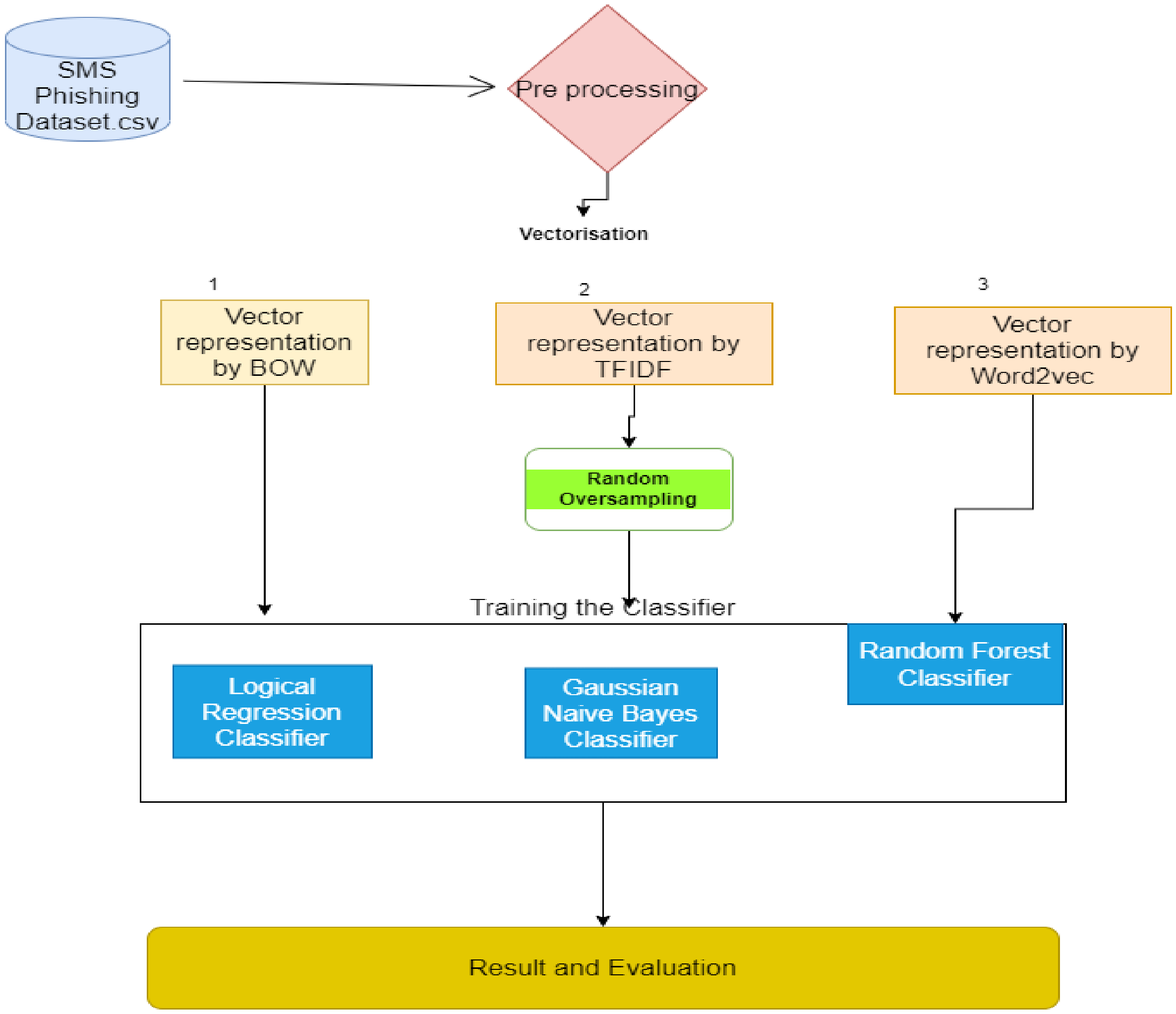

SMS Phishing Dataset.csv
Pre processing
Vectorisation
1
Vector representation by BOW
2
Vector representation by TFIDF
3
Vector representation by Word2vec
Random Oversampling
Training the Classifier
Logical Regression Classifier
Gaussian Naive Bayes Classifier
Random Forest Classifier
Result and Evaluation

Fig 1 Flowchart of Natural language Processing on SMS Phishing dataset using Machine Learning Models

3.1 Dataset

The "SMS phishing dataset for machine learning and pattern recognition" used for the project was open source and available on Mendeley data contributed by Mishra S. and Soni D. (2022). It had a collection of labelled text messages used in SMS phishing research. It contained 5971 texts messages labelled as Legitimate /Ham (4844), Spam (489), and Smishing (638) as detailed in Table 1.

Label	Number	Percentage
Legitimate (Ham)	4844	81.8
Spam	489	8.1
Smishing	638	10.6

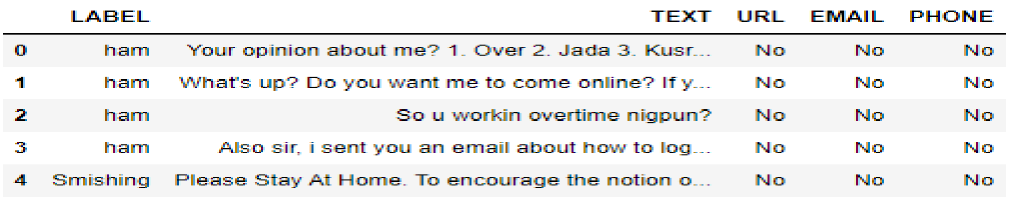
Table 1: Original Dataset label

As per our objective for the research, we need to classify the messages as phishing or no phishing, thus we regrouped the labels. Smishing label had been renamed as Phishing, Spam and Ham together as no phishing as shown in Table 2.

Label	Number	Percentage
NO Phishing (Ham + Spam)	5333	89.3
Phishing (Smishing)	638	10.6

Table 2: Dataset Label for the research

The dataset obtained for the research had five different columns of properties respectively as Label, Text, URL, EMAIL and PHONE and 5971 rows of messages. URL, EMAIL and PHONE columns details about presence or absence of these in the Text of messages. Fig 2 presents a sample of raw dataset with five different columns and a few rows.



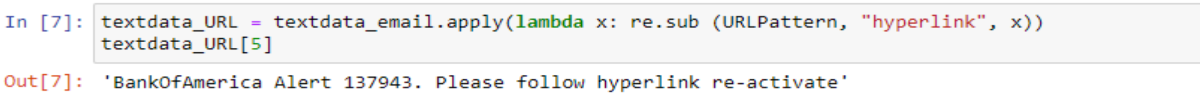
	LABEL	TEXT	URL	EMAIL	PHONE
0	ham	Your opinion about me? 1. Over 2. Jada 3. Kusr...	No	No	No
1	ham	What's up? Do you want me to come online? If y...	No	No	No
2	ham	So u workin overtime nigpun?	No	No	No
3	ham	Also sir, i sent you an email about how to log...	No	No	No
4	Smishing	Please Stay At Home. To encourage the notion o...	No	No	No

Fig 2 Raw dataset

3.2 Data Pre-processing

Dataset to be worked upon need to be processed as the messages may contain information irrelevant for the analysis but if present during the process would have led towards noise and large size. Pre-processing of data presented better features for analysis, avoid consumption of unnecessary resources and delay in the result. Different steps of pre-processing used were as follows:

- **Replacing the mobile number, e-mail, and URL pattern –** The mobile number, email, and URL pattern, if present in the text had been replaced with the keywords mobile number, e-mail, and hyperlink respectively for the purpose of cleaning and a better understanding of dataset.

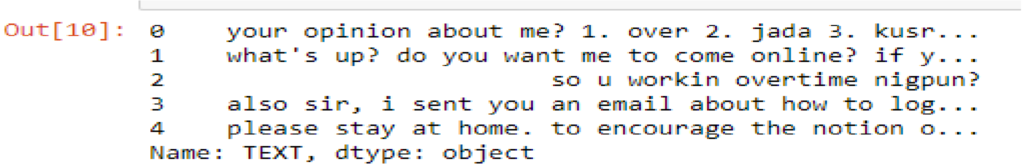


```
In [7]: textdata_URL = textdata_email.apply(lambda x: re.sub (URLPattern, "hyperlink", x))
        textdata_URL[5]

Out[7]: 'BankOfAmerica Alert 137943. Please follow hyperlink re-activate'
```

Fig 3 : Replacing email pattern in text with keyword hyperlink

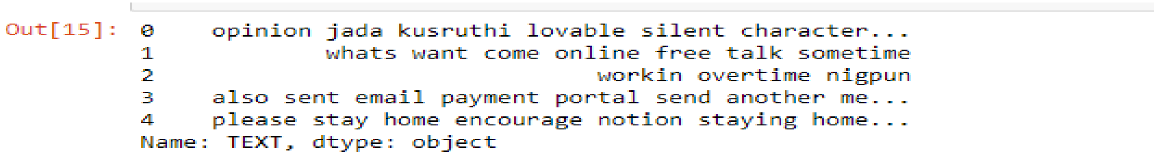
- **Converting the text to lower case**- The computer could identify the same word in lower and capital letters as different words, leading to confusion during the training. Thus, the text column of the dataset had been converted to lower case, Fig 4 presents TEXT column of dataset in lower case.



```
Out[10]: 0    your opinion about me? 1. over 2. jada 3. kusr...
         1    what's up? do you want me to come online? if y...
         2                             so u workin overtime nigpun?
         3    also sir, i sent you an email about how to log...
         4    please stay at home. to encourage the notion o...
         Name: TEXT, dtype: object
```

Fig 4 TEXT column of dataset in lower case.

- **Removing special character-**Special characters present in the text like (?   &, /, @) could accounts to noise so had been removed during pre-processing**.**
- **Removing numeric character-**Any numeric value present in the text had been removed as it would have added to the noise only.
- **Word Tokenization**- Tokenization is the division of text into words and sentences. The unstructured data could be converted into structured with tokenization. Here we did use word tokenizer as the aim of the research, to identify the luring, compelling words leading to phishing attack through messages.
- **Removing short word**-The text data contained short words like a, an, spl which did not make sense in understanding the natural language for the purpose and thus had been removed of length (1-3).
- **Stop word removal**- Different stop words like the, an, a, had been removed from the text. Figure 5 present few rows of pre-processed TEXT column from dataset.



```
Out[15]: 0    opinion jada kusruthi lovable silent character...
         1             whats want come online free talk sometime
         2                               workin overtime nigpun
         3    also sent email payment portal send another me...
         4    please stay home encourage notion staying home...
         Name: TEXT, dtype: object
```

Fig 5 Pre-processed TEXT Column of dataset

3.3 Vectorisation

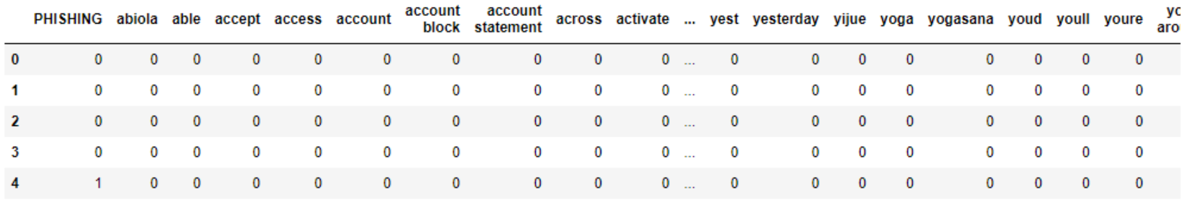
Vectorization turns human-written material into comprehensible numerical representations or machine-readable forms. In the proposed research we had used three different methods of vectorization to represent the pre-processed data for machine learning. The aim of using three different methods is to compare and analyse the best one to achieve the aim of detection of phishing in instant messages correctly. In addition to the vectorization techniques used for representation of text of messages, we did use hot encoding for vector representation of labels in the dataset.

Encoding the categorial column- To our project, the LABEL of dataset had been replaced with PHISHING and regrouped further where, Smishing = Phishing and Ham + Spam= no phishing**.** The categorical dependent column that is Phishing or Legitimate had been encoded for the purpose of classification as Phishing = 1 and no phishing = 0.

Data transformation/vectorization

Machine learning models needed data to be processed in form of numbers as machine could not understand and interpret the textual information unless presented in meaningful numeric representation. The following three techniques had been used for the creation of the representation of textual data of messages in the form of numbers (machine readable form):

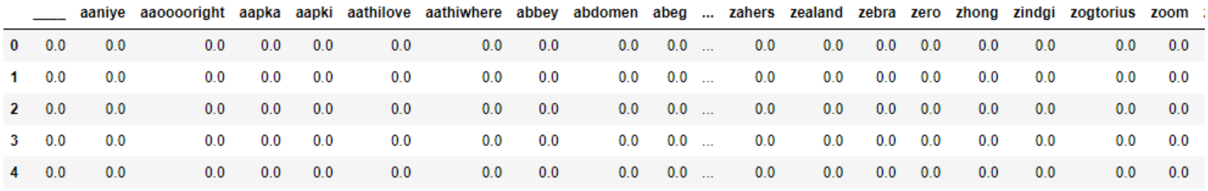
Bag of Words (BOW) It is the simplest model for representation of words in vector form. Here the frequency of repeated words is considered and does not count the order of words in document. Length of vectors keeps on increasing with each new word being added in the dataset and the matrix looks sparse being 0s being added if any word is absent in that document. There is no order of words in the text being followed in the matrix which could lead to hinderance of meaning. The Python Count Vectorizer library was used to apply BOW. N-grams can be used to implement the Count Vectorizer. Utilizing a unigram and bigram matrix, the feature was retrieved for the study. The Bigram matrix consisted of two consecutive words from a document, while the Unigram matrix featured one word. Fig 6 represents few rows of vectorized dataset using BOW.



	PHISHING	abiola	able	accept	access	account	account block	account statement	across	activate	...	yest	yesterday	yijue	yoga	yogasana	youd	youll	youre	yc aro
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

Fig 6 Vector representation using Bag of words

Term Frequency-Inverse Document Frequency (TF-IDF) is a typical technique for converting text data into a vector format. It stands for "Term Frequency-Inverse Document Frequency." This matrix aids in comprehending the word's significance throughout the corpus of documents. Here two things are calculated that is Term frequency that is no. of time a word appears in the document and Inverse Document Frequency that means log of number of documents divided by number of documents containing that word. Hence, the more frequent words used in document were weighted low in terms of vector representation, but less frequent and important words had high IDF value as vector representation (Bountakas P. et. al. 2022). The vector representation of document using TF-IDF are presented by Fig 7.



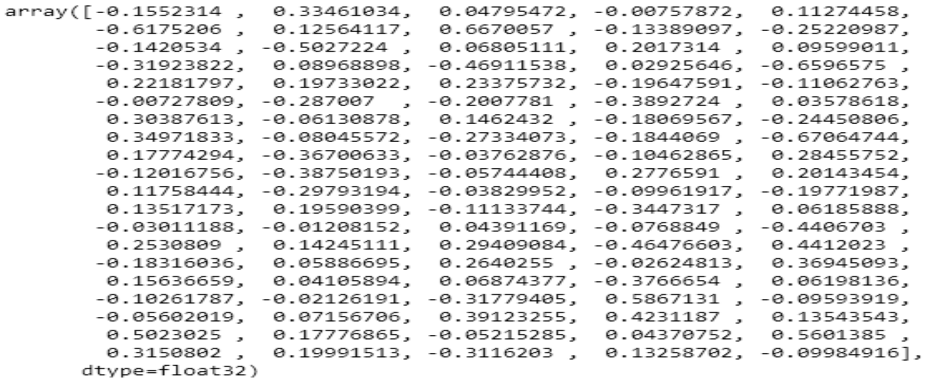
	____	aaniye	aaoooright	aapka	aapki	aathilove	aathiwhere	abbey	abdomen	abeg	...	zahers	zealand	zebra	zero	zhong	zindgi	zogtorius	zoom
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig 7 Vector representation using TF-IDF

Word2vec Embedding

Word2vec is a technique to represent the words in dense vector form in space where the words related in contexts have similar vectors. This is an advantage of using word2vec in detection of phishing as attackers keep modifying the language to avoid the direct bait of phishing. With the help of this tool, it is possible to turn words into vectors, calculate their distances, and construct an analogy between the words. Word2vec can identify the semantic connections between the words in a sentence (Al-Saqqa & Awajan, 2019).

Word2Vec builds word vectors, which are distributed numerical representations of word features. These word features may include words that indicate the context of the specific vocabulary words that are present individually. Through the generated vectors, word embeddings eventually assist in forming the relationship of a word with another word having a similar meaning. Gensim library had been used in research to create the word embedding of dataset and a few arrays of transformed vector had been presented in Fig 8.



```
array([-0.1552314 ,  0.33461034,  0.04795472, -0.00757872,  0.11274458,
       -0.6175206 ,  0.12564117,  0.6670057 , -0.13389097, -0.25220987,
       -0.1420534 , -0.5027224 ,  0.06805111,  0.2017314 ,  0.09599011,
       -0.31923822,  0.08968898, -0.46911538,  0.02925646, -0.6596575 ,
        0.22181797,  0.19733022,  0.23375732, -0.19647591, -0.11062763,
       -0.00727809, -0.287007  , -0.2007781 , -0.3892724 ,  0.03578618,
        0.30387613, -0.06130878,  0.1462432 , -0.18069567, -0.24450806,
        0.34971833, -0.08045572, -0.27334073, -0.1844069 , -0.67064744,
        0.17774294, -0.36700633, -0.03762876, -0.10462865,  0.28455752,
       -0.12016756, -0.38750193, -0.05744408,  0.2776591 ,  0.20143454,
        0.11758444, -0.29793194, -0.03829952, -0.09961917, -0.19771987,
        0.13517173,  0.19590399, -0.11133744, -0.3447317 ,  0.06185888,
       -0.03011188, -0.01208152,  0.04391169, -0.0768849 , -0.4406703 ,
        0.2530809 ,  0.14245111,  0.29409084, -0.46476603,  0.4412023 ,
       -0.18316036,  0.05886695,  0.2640255 , -0.02624813,  0.36945093,
        0.15636659,  0.04105894,  0.06874377, -0.3766654 ,  0.06198136,
       -0.10261787, -0.02126191, -0.31779405,  0.5867131 , -0.09593919,
       -0.05602019,  0.07156706,  0.39123255,  0.4231187 ,  0.13543543,
        0.5023025 ,  0.17776865, -0.05215285,  0.04370752,  0.5601385 ,
        0.3150802 ,  0.19991513, -0.3116203 ,  0.13258702, -0.09984916],
      dtype=float32)
```

Fig 8 Vector representation using Word2vec

3.4 Classification Model

The text data from instant messages needed to be classified into phishing and no phishing as per the aim of the research method. We employed three different classifiers for classification and detection to compare the findings and determine which was better for the task. Three different supervised machine learning model used in this study were: Random Forest, Gaussian Naive Bayes, and Logistic Regression. These machine learning models were selected because of their performance and frequent use by the research community. The first stage in creating a classification model was to train the chosen model using training data and then test the model with test data. Train test split of 0.33 was used to split the vectorised data for the purpose of training the models and evaluating it.

- **Logistic Regression Classifier**

Logistic regression classifier is based on the principle of linear regression but since the output is classifying the data into binary, hence fits the classification model. Here, Regression is subjected to a logistic function to determine the likelihood that it belongs to either class. It compares the log of the likelihood that an event will occur to the log of the likelihood that it will not. In the end, it categorizes the variable according to which class has the larger likelihood (Nitish Sharma,2022).

- **Naïve Bayes Classifier**

A straightforward probability technique called Naive Bayes is based on the idea that each feature of the model is independent of the others. In the context of the phishing filter, we assume that each word in the message is distinct from every other word, and we tally the words without taking context into account (A. Kovač, et.al. 2022).

- **Random Forest Classifier**

One of the most popular machine learning techniques for classification and regression is random forests. A supervised machine learning system called a random forest combines the results of many decision trees' calculations to produce a single outcome (A. Kovač, et.al. 2022). It is well-liked because it is straightforward yet efficient. Random forests are based on the idea that while each tree might be rather good at predicting, it will almost certainly overfit on some data. They typically work well without a lot of parameters changing, are quite powerful, and do not require data scalability.

3.5 Cross validation and Evaluation

The machine learning models could be biased if train and test data is not utilised properly, so we did imply the stratified K Fold cross validation. Stratified K-Fold cross validation helped to create no. of K Fold in a manner that both classification label will be randomly distributed but proportionally in all the folds of split used for training and testing of classifier. We did 10-fold cross validation for our research.

Further parameters of individual classifiers were tuned to improve the specificity and efficiency of models.

4 Design Specification

The aim of the research is to process the message using NLP to find the keywords in messages that are explicitly used to lure the users to divulge private information. This could help the machine learning model to help to classify the messages as phishing messages or no phishing message. The output of the model is to predict the message as phishing or no phishing, depending on the contextual meaning of the words in the messages either alone or together with the presence of malicious link, e-mail, and mobile number. Fig 9 represents the proposed/ future model of Phishing detector in Mobile Instant Messaging with following components. This research had focused on Instant Message analyzer using NLP and ML, which can be combined with other proposed components of the future design to shape the model presented in Fig 9.

Phishing detector model in Instant Messenger

1. Malicious mobile number analyzer
2. Malicious URL link analyzer
3. Malicious E-mail analyzer
4. Instant Message analyzer using NLP and ML

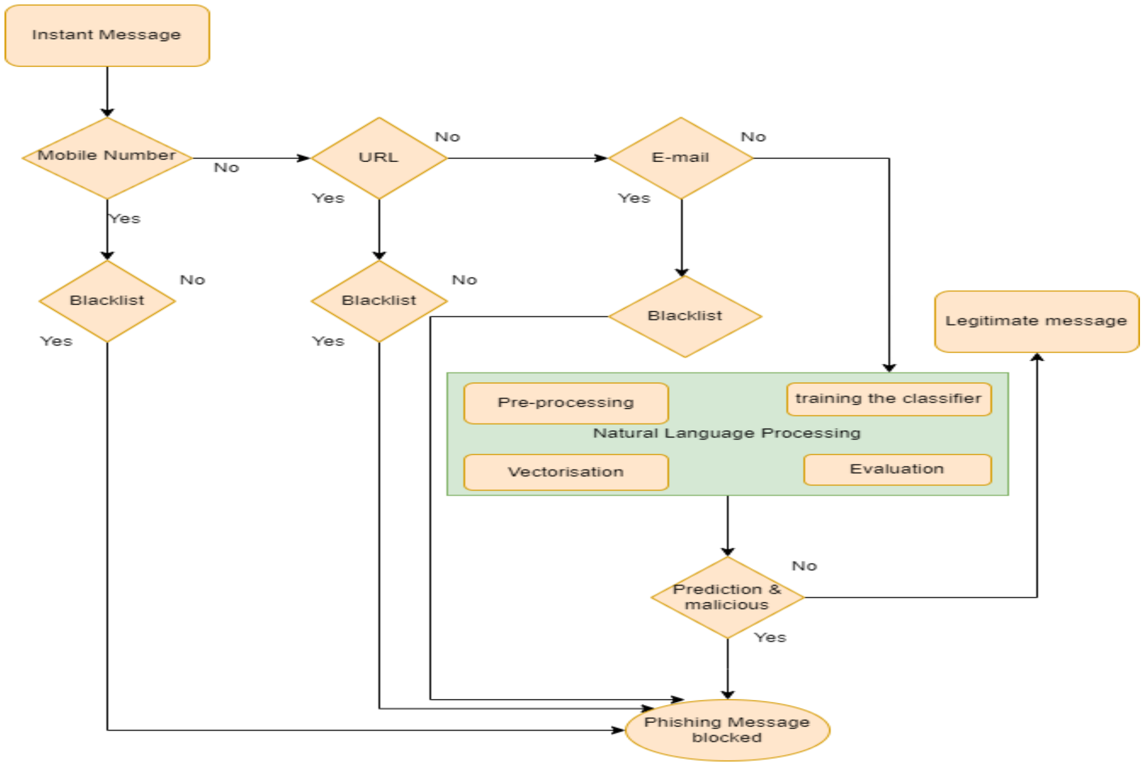


Fig 9 Proposed flowchart of model for phishing detection in Mobile Instant Messaging

4.1 Malicious mobile number analyser

The text messages could be sent from a blacklist mobile number or could contain a blacklist number for further communication. In this situation, the mobile number will be checked for blacklist and if it matches, the model will flag it as phishing message and blocked.

4.2 Malicious URL link analyser

The text messages in instant messages might contain a malicious link, thus if any link is present in the messages, it would be needed to check the malicious link if it was a blacklist or not. If the link match with the blacklist, the model would flag the message as phishing message and blocked.

4.3 Malicious E-mail analyser

The text messages could contain a blacklist e-mail for further communication. If the e-mail exists in blacklist, the model will flag it as phishing message and blocked.

4.4 Instant Message analyser using NLP and ML

The messages in instant messages could be in conversation form and the attackers invest time in building the trust through earlier conversation to initiate phishing attack. With all knowledge and communication about phishing attacks, the attackers keep inventing new strategy for attacks, which could be build up of story before the attack. This drives the need for message analysis utilizing natural language processing (NLP) to comprehend the context of words in messages and predict the aim of phishing to classify and stop phishing messages before they reach consumers.

In the current research we focussed on this aspect of instant messages, where the text would be analysed for presence of malicious keywords aiming to phishing. The language is tweaked according to the user and the continuing conversation to sensitize the target in a state of mind to act as per the attacker. Social engineers no longer ask direct questions to divulge confidential responses or use commands to trigger any action.

In this research we have experimented with different vectorization methods and different classifiers to extract the best features and analyse to accurately classify the instant messages as phishing and no phishing.

5 Implementation

To practically demonstrate the objective of the research by analysis of instant message using NLP and ML, the methodological steps as explained above were implemented in the Jupyter notebook (Anaconda) as developing and learning for the research using Python as programming language.

5.1 Prerequisites

On Windows 11, we installed the open-source Anaconda Distribution and utilized Jupyter Notebook version 6.4.8 as the development and learning environment for our project and machine learning model creation.

- Programming language: Python 3.9.12
- Jupyter notebook version 6.4.8
- Operating System- Windows 11

5.2 Dataset

We used open-source dataset from Medley data “SMS phishing dataset for machine learning and pattern recognition”. We imported pandas as pd to read the dataset in Jupyter and conduct various operations.

- Pandas- It is a library in python language for data analysis.

5.3 Pre-processing

Pre-processing of data influences the machine learning approach and help in proper selection of features for the purpose of analysis. Steps of pre-processing has been explained in earlier section but to perform this step we required following libraries in the developing environment:

- NumPy- NumPy is used for mathematical operations.
- NLTK (Natural Language Toolkit) is the framework to build python program in NLP. It has inbuilt libraries for tokenization, lemmatization, stemming, stop word removal.

5.4 Machine Learning

- Scikit-learn/sklearn library enabled to implement machine learning in Python and allowed us to conduct a variety of machine learning tasks like feature extraction using CountVectorizer, TfidfVectorizer, train_test_split.
- sklearn enabled to apply various classification models like GaussianNB, Logical Regression and RandomForest.
- Metrics for evaluation like confusion_matrix, classification_report. roc_curve could be done using sklearn.
- Gensim – It offers topic modelling, document indexing, and similarity retrieval with huge corpora in python.

5.5 Data balancing

The dataset utilised for the analysis was imbalanced with 9:1 ratio of Legitimate and phishing message so we did implied ROS (Random Over Sampling) on our X train and Y train set of data. imblearn library was used for Random Over Sampling.

5.6 k-Fold cross validation

Machine learning algorithm needed to be tested for accuracy through cross-validation to avoid over-fitting on dataset. Here we had employed 10-fold cross validation by analysing the model to be cross validated 10 times by splitting the dataset into 10 groups and training and testing on these groups.

- Sklearn library was used to import cross_val_score.

5.7 Data visualisation

- matplotlib library had been used for visualising the data for evaluation.

5.8 Parameter tuning

Parameter tuning helped to get the best and consistent result for the analysis.

- n_estimator with cross_val_score was used to tune the parameter for Random Forest model.

6 Evaluation and Discussion

To detect and isolate the phishing messages in instant messaging application, we trained machine learning classifiers with three different feature extraction method Bag of words (BOW), Term Frequency Inverse Document Frequency (TFIDF) and word2vec. The evaluation of different models' performance was based on the following metrics of namely: Accuracy, Precision, Recall, F1- score., Receiver Operating Characteristics (ROC) and Area under the ROC Curve (AUC).

6.1 Metrics

- Accuracy is calculated as the proportion of accurate predictions to all other predictions.

$$\textbf{Accuracy} = \frac{\textbf{Number of correct predictions}}{\textbf{Total number of predictions}}$$

- **Number of correct predictions** = TP+TN
- **Total number of predictions** = TP+TN+FP+FN

- Precision determines the fraction of positive predictions that were accurate.

$$\textbf{Precision} = \textbf{TP /(TP+FP)}$$

- Recall seeks to determine the percentage of actual positives that were mistakenly detected.

$$\textbf{Recall (TPR)} = \textbf{TP/(TP+FN)}$$

- F1 Score: A binary classification model is evaluated using the F1 Score metric based on the predictions provided for the positive class. With the use of Precision and Recall, it is calculated. It is a specific kind of score that combines Precision and Recall.

$$\textbf{F1 score} = 2 * \frac{\textbf{Precision * recall}}{\textbf{Precision+ recall}}$$

- Receiver Operating Characteristics (ROC): The Receiver Operating Characteristics graph displays the TPR (Total Positive Rate) and the FPR (False Positive Rate) (ROC). It explains explicitly how a classification algorithm functions at various categorization criteria. The ROC curve should, as much as feasible, resemble an upside-down L, according to a straightforward rule of thumb.
- Area under the ROC Curve (AUC): It measures the area under the ROC curve.

For the objective of our research and to evaluate the performance of model suited for further research, we focussed our evaluation metrics on the accuracy of the model and the recall for phishing messages (phishing= 1). The implementation of research framework had been as per research methodology and evaluation of NLP techniques and models were as followed:

6.2 Experimental Set up 1: Bag of word Vectorization Technique

6.2.1 Applied to Logical Regression Classifier

BoW is the method of vectorisation which have been utilised here to train the classifiers. We started with the simplest classifier, Logical regression to build our classification model of phishing detection. Table 3 presents the performance metrics of Logical Regression classifier as per classification report of model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 3 below.

Accuracy	Precision	Recall	F 1 score
0.97	0.91	0.79	0.84

Table 3 Performance metrics of Logical Regression Classifier for Phishing messages = 1

6.2.2 Applied to Gaussian Naïve Bayes Classifier

The Gaussian Naive Bayes Classifier was developed and tested as the second classifier model to be trained. Table 4 presents the performance metrics of Gaussian Naïve Bayes Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 4 below.

Accuracy	Precision	Recall	F 1 score
0.84	0.38	0.88	0.54

Table 4 Performance metrics of Gaussian Naïve Bayes Classifier for Phishing messages = 1

6.2.3 Applied to Random Forest Classifier

Table 5 presents the performance metrics of Random Forest Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 5 below.

Accuracy	Precision	Recall	F 1 score
0.97	0.90	0.79	0.85

Table 5 Performance metrics of Random Forest Classifier for Phishing messages = 1

6.2.4 **Cross validation of Models with Stratified K Fold validation**

The number of K Folds was increased with the use of stratified K-fold cross validation, ensuring that both categorization labels would be distributed randomly but proportionately throughout all folds of the split. All the models as trained above and tested for the dataset had been cross validated with Stratified K Fold having k =10.

Logical Regression	Random Forest	Gaussian Naïve Bayes
0.970	0.971	0.840

Table 6 Accuracy of models with BoW upon Cross validation

6.2.5 Parameter tuning of Random Forest Classifier

As evident Random Classifier outperformed the other two classifiers in terms of accuracy and recall, therefore, we did further fine tune the parameter (n_estimator) in Random Forest Classifier. When **n_ estimators =20**, the model performed with highest average value of **0.9706**.

6.3 Experiment Set up 2: Term Frequency-Inverse Document Frequency Vectorization Technique

Further, we vectorized the dataset with TFIDF (Term Frequency Inverse Document Frequency) as a method for vectorization as it gives more weightage to less used term in document which could be the better way to select features for training and testing in our case. To balance the dataset, we used ROS (Random Over Sampling) on the X train and Y train sets of data.

The document matrix obtained is further used to train and test different classifier to get better understanding of analysis.

6.3.1 Applied to Logical Regression Classifier

The logical regression model was trained using the document matrix produced by TFIDF, and the results showed 97% accuracy in classifying instant messages as phishing and legitimate, with 92% recall for phishing messages. Table 7 presents the performance metrics of Logical Regression Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 7 below.

Accuracy	Precision	Recall	F 1 score
0.97	0.80	0.91	0.85

Table 7 Performance metrics of Logical Regression Classifier for Phishing messages = 1

6.3.2 Applied to Gaussian Naïve Bayes Classifier

Gaussian Naïve Bayes Classifier presented the accuracy of 87% while classifying the messages into two labels with 77% recall for phishing messages**.** Table 8 presents the performance metrics of Gaussian Naïve Bayes Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 8 below.

Accuracy	Precision	Recall	F 1 score
0.87	0.45	0.77	0.57

Table 8 Performance metrics of Gaussian Naïve Bayes Classifier for Phishing messages = 1

6.3.3 Applied to Random Forest Classifier

Random Forest Classifier presented the accuracy of 97% while classifying the messages into two labels with 85% recall for phishing messages**.** Table 8 presents the performance metrics of Random Forest Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 9 below.

Accuracy	Precision	Recall	F 1 score
0.97	0.89	0.84	0.87

Table 9 Performance metrics of Random Forest Classifier for Phishing messages = 1

6.3.4 Cross validation of Models with Stratified K Fold validation

The number of K Folds was increased with the use of stratified K-fold cross validation, ensuring that both categorization labels would be distributed randomly but proportionately throughout all folds of the split. All the models as trained above and tested for the dataset had been cross validated with Stratified K Fold having k =10. Accuracy scores after cross validation of models had been presented in Table 10.

Logical Regression	Random Forest	Gaussian Naïve Bayes
0.982	0.992	0.935

Table 10 Accuracy of models with TFIDF on upon Cross validation

As evident from the scores Random Forest Classifier and Logical Regression model presented with similar accuracy on dataset when used TFIDF vectorisation and Random oversampling. Once the models were cross validated with K-Fold, Random Foreset outperformed the Logical regression model.

6.3.5 Parameter tuning with Random Forest Classifier

As evident Random Classifier outperformed the other two classifiers, therefore, we did further fine tune the parameter(n_estimator) in Random Forest Classifier. When **n_ estimators =40**, the model performed with maximum average accuracy of detection and classification of **0.992**.

6.4 Experiment Set up 3: Word2vec as vectorization technique

We used the Word2vec word embedding method from the Gensim package. Word2vec is a vector representation of the dataset that considers the context of the words used in a message. As a result, Word2vec assigns similar vector scores to words with similar context. This causes training of the classifier with labels accurately and thus enhanced prediction.

6.4.1 Applied to Random Forest Classifier

Random Forest Classifier responded well and outperformed other two models, so the vectorised data produced by Word2vec was trained and tested only on Random Forest Classifier. Table 11 presents the performance metrics of Random Forest Classifier as per classification report of the model. The precision, recall and F1 score of phishing message(phishing=1) were presented in table 11 below.

Accuracy	Precision	Recall	F 1 score
0.95	0.87	0.66	0.75

Table 11 Performance metrics of Random Forest Classifier with Word2vec

The Random forest classifier was further cross validated with 10 fold stratified cross validation and presented the accuracy of 95%. On tuning the n_estimator parameter of RF, the average accuracy score was 95.2% with Random Forest and Word2vec.

7 Result

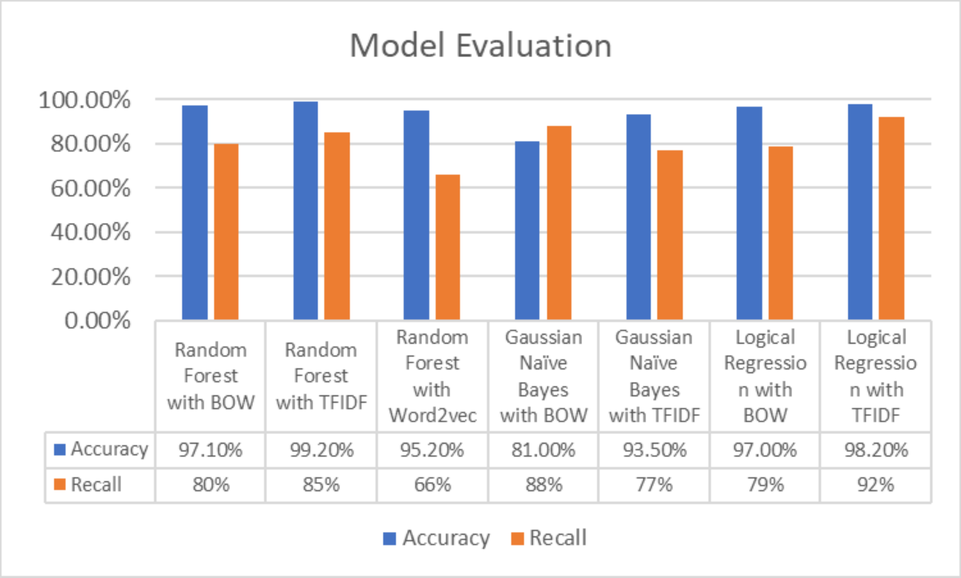
Three different combination of NLP techniques (BOW, TF-IDF and Word2vec) and ML models (Logical regression, Gaussian Naïve Bayes, and Random Forest) have been experimented and evaluated for the research. The effectiveness of the classification model was assessed based on recall of phishing messages and accuracy of prediction of the right labels of legitimate and no phishing messages in instant messaging.

All the machine learning algorithm performed well with an accuracy of phishing detection over 84% in case of Bag of Word vectorisation as explained in **Table 6 Accuracy of models with BOW upon Cross validation.** In TFIDF as vectorization method, it weighted the less frequent words more and thus proved to be better feature extraction technique for training the classifier. The train dataset was further balanced to avoid overfitting. The models trained with TFIDF when cross validated with Stratified K- Fold showed improved performance, where Random Forest outperformed others with 99.2 accuracy, as depicted in **Table 10 Accuracy of models with TFIDF on upon Cross validation.** With Word2vec word embedding, only Random Forest classifier was trained and tested due to its consistent performance. Here the Random Forest presented the accuracy of 95.2% upon cross validation.

Hereby we the evaluated the performance of three supervised models with three different techniques of vectorization as represented in Table 12. The table contrasts the matrix of evaluation to provide the most effective model and approach. Fig 10 presents the bar chart for model Evaluation based on Accuracy and Recall in (%) of different models using different vectorization techniques.

Model / Vectorisation	Random Forest		Logical Regression		Gaussian Naïve Bayes	
	Accuracy	**Recall**	**Accuracy**	**Recall**	**Accuracy**	**Recall**
TFIDF with ROS data	0.992	85	0.982	92	0.935	77
Word2vec	0.952	66	NA		NA	
BoW	0.971	80	0.970	79	0.81	88

Table 12 Performance metrics of Classifiers



	Random Forest with BOW	Random Forest with TFIDF	Random Forest with Word2vec	Gaussian Naïve Bayes with BOW	Gaussian Naïve Bayes with TFIDF	Logical Regressio n with BOW	Logical Regressio n with TFIDF
■ Accuracy	97.10%	99.20%	95.20%	81.00%	93.50%	97.00%	98.20%
■ Recall	80%	85%	66%	88%	77%	79%	92%

Fig 10 Model Evaluation based on Accuracy and Recall in (%) of different models using different vectorization techniques.

8 Conclusion and Future Work

This research presents a phishing detection and classification approach in instant messaging using natural language processing and machine learning algorithm. By contrasting several vectorization techniques and machine learning (ML) models for the detection of phishing instant messages, we developed a method to build the model. The BOW, TF-IDF and Word2vec methods of vectorization using NLP were utilised to create document matrix from messages. Each document matrix created were trained and tested on Logical Regression, Gaussian Naïve Bayes, and Random Forest classifiers for the purpose of classifying messages into phishing or no phishing messages. When the dataset was balanced with ROS and vectorized using TFIDF, Random Forest Classifier outperformed other models with accuracy for phishing detection of 99.2% and recall of 85% for phishing messages.

It is evident that less emphasis has been placed on research involving phishing in Instant Messages and the lack of a well-defined dataset for Instant Messages can be a hinderance towards our goal of using Natural language Processing to detect phishing in Instant Messages using contextual relationships between words and sentences. This could be remedied by developing a specific dataset and having society participate by updating it frequently. As a future work, we will focus on developing dataset specific for phishing in instant messages. Additionally, as a proposed design, this research can be combined as a model with the ability to detect likely phishing messages from fraudulent mobile numbers, containing malicious email addresses and malicious URLs/executable files on a large scale, with organizations having Instant Messaging applications to launch it on a commercial scale.

Integration of Word2vec in phishing detection looks promising thus providing fuel for further research in the area.

Additionally, the system and data collection should be updated frequently as the phishing techniques keeps changing with use of different tools. Integration of NLP for classification using different machine learning techniques is an area of Data Analytics which is a vast field and can be utilised for new research. This approach will help people avoid falling for social engineering through instant messages in their daily lives if it is incorporated into instant messaging software.

References

(2022) *PHISHING ACTIVITY TRENDS REPORT, 2nd Quarter 2022*. publication. Available at: https://apwg.org/trendsreports/ (Accessed: October 9, 2022).

Ahmad, R., Terzis, S. (2022). Understanding Phishing in Mobile Instant Messaging: A Study into User Behaviour Toward Shared Links. In: Clarke, N., Furnell, S. (eds) Human Aspects of Information Security and Assurance. HAISA 2022. IFIP Advances in Information and Communication Technology, vol 658. Springer, Cham. https://doi.org/10.1007/978-3-031-12172-2_15

Scheeres, J., 2008. *Establishing the Human Firewall: Reducing an Individual's Vulnerability to Social Engineering Attacks*. [online] apps.dtic.mil. Available at: <https://apps.dtic.mil/sti/citations/ADA487118> [Accessed 5 October 2022].

Salahdine, F. and Kaabouch, N. (2019) “Social Engineering Attacks: A survey,” *Future Internet*, 11(4), p. 89. Available at: https://doi.org/10.3390/fi11040089.

Jain, A.K., Debnath, N. and Jain, A.K. (2022) “APUML: An efficient approach to detect mobile phishing webpages using machine learning,” *Wireless Personal Communications*, 125(4), pp. 3227–3248. Available at: https://doi.org/10.1007/s11277-022-09707-w.

Stone, A., 2007. Natural-Language Processing for Intrusion Detection. *Computer*, 40(12), pp.103-105.
Mishra, s. and Soni, D., 2022. *SMS PHISHING DATASET FOR MACHINE LEARNING AND PATTERN RECOGNITION*. [online] Mendeley Data. Available at: <https://data.mendeley.com/datasets/f45bkkt8pr/1> [Accessed 15 October 2022].

Alabdan, R., 2020. Phishing Attacks Survey: Types, Vectors, and Technical Approaches. *Future Internet*, 12(10), p.168.

Boateng, E.O.Y. and Amanor, P.M. (2014) “Phishing, SMiShing & Vishing: An Assessment of Threats against Mobile Devices,” *Journal of Emerging Trends in Computing and Information Sciences*, 5.

Guan, D., Chen, C. and Jia-Bin, L., 2022. Anomaly Based Malicious URL Detection in Instant Messaging. *ResearchGate*, [online] Available at: <https://www.researchgate.net/publication/267221007_Anomaly_Based_Malicious_URL_Detection_in_Instant_Messaging> [Accessed 1 November 2022].

C. Singh and Meenu, "Phishing Website Detection Based on Machine Learning: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 398-404, doi: 10.1109/ICACCS48705.2020.9074400.

Ali, M. and Rajamani, L., 2022. *Global Trends in Computing and Communication Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.490-502.

Y. Sawa, R. Bhakta, I. G. Harris, and C. Hadnagy, "Detection of Social Engineering Attacks Through Natural Language Processing of Conversations," 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), 2016, pp. 262-265, doi: 10.1109/ICSC.2016.95.

Peng, T., Harris, I. and Sawa, Y., 2018. Detecting Phishing Attacks Using Natural Language Processing and Machine Learning. In: *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. IEEE.

Mishra, S. and Soni, D., 2020. Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis. *Future Generation Computer Systems*, 108, pp.803-815.

Lopez, J.C., and Camargo, J.E., 2022, March. Social Engineering Detection Using Natural Language Processing and Machine Learning. In *2022 5th International Conference on Information and Computer Technologies (ICICT)* (pp. 177-181). IEEE.

Mishra, Sandhya; Soni, Devpriya (2022), "SMS PHISHING DATASET FOR MACHINE LEARNING AND PATTERN RECOGNITION", Mendeley Data, V1, doi: 10.17632/f45bkkt8pr.1

A. Kovač, I. Dunđer and S. Seljan, "An overview of machine learning algorithms for detecting phishing attacks on electronic messaging services," 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), 2022, pp. 954-961, doi: 10.23919/MIPRO55190.2022.9803517.

N. Sharma, "A Methodological Study of SMS Spam Classification Using Machine Learning Algorithms," 2022 2nd International Conference on Intelligent Technologies (CONIT), 2022, pp. 1-5, doi: 10.1109/CONIT55038.2022.9848171.

Bountakas, P., Koutroumpouchos, K. and Xenakis, C. (2021) "A comparison of natural language processing and machine learning methods for phishing email detection," *The 16th International Conference on Availability, Reliability and Security* [Preprint]. Available at: https://doi.org/10.1145/3465481.3469205.

Y. Lan, "Chat-Oriented Social Engineering Attack Detection Using Attention-based Bi-LSTM and CNN," 2021 2nd International Conference on Computing and Data Science (CDS), 2021, pp. 483-487, doi: 10.1109/CDS52072.2021.00089.

Al-Saqqa, S. and Awajan, A. (2019) "The use of word2vec model in sentiment analysis," *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control* [Preprint]. Available at: https://doi.org/10.1145/3388218.3388229.