Sentence Level Analysis Model for Phishing Detection

Preprint · September 2023

DOI: 10.22541/au.169445734.42206568/v1

4 authors, including:

Lindah Sawe
Mount Kenya University
2 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Joyce W. Gikandi
Mount Kenya University
38 PUBLICATIONS **1,663** CITATIONS
SEE PROFILE

David Njuguna
MURANG'A UNIVERSITY OF TECHNOLOGY
10 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Journal of Cyber Security

Tech Science Press

DOI: 10.32604/jcs.2023.045859

ARTICLE

Sentence Level Analysis Model for Phishing Detection Using KNN

Lindah Sawe*, Joyce Gikandi, John Kamau and David Njuguna

Faculty of Computing and Informatics, Mount Kenya University, Thika, Kenya

*Corresponding Author: Lindah Sawe. Email: lynsawe@gmail.com, lsawe@mku.ac.ke

Received: 10 September 2023 Accepted: 23 November 2023 Published: 11 January 2024

ABSTRACT

Phishing emails have experienced a rapid surge in cyber threats globally, especially following the emergence of the COVID-19 pandemic. This form of attack has led to substantial financial losses for numerous organizations. Although various models have been constructed to differentiate legitimate emails from phishing attempts, attackers continuously employ novel strategies to manipulate their targets into falling victim to their schemes. This form of attack has led to substantial financial losses for numerous organizations. While efforts are ongoing to create phishing detection models, their current level of accuracy and speed in identifying phishing emails is less than satisfactory. Additionally, there has been a concerning rise in the frequency of phished emails recently. Consequently, there is a pressing need for more efficient and high-performing phishing detection models to mitigate the adverse impact of such fraudulent messages. In the context of this research, a comprehensive analysis is conducted on both components of an email message—namely, the email header and body. Sentence-level characteristics are extracted and leveraged in the construction of a new phishing detection model. This model utilizes K Nearest Neighbor (KNN) introducing the novel dimension of sentence-level analysis. Established datasets from Kaggle were employed to train and validate the model. The evaluation of this model's effectiveness relies on key performance metrics including accuracy of 0.97, precision, recall, and F1-measure.

KEYWORDS

Sentence level analysis; email header; email body; phishing detection; KNN

1 Introduction

Email attacks stand out as one of the most prevalent threats that everyday internet users encounter both at work and at home. Emails serve as a widespread communication medium, with over 70% of surveyed individuals relying on them for remote work and connecting with friends and colleagues. However, many individuals remain unaware of the potential risks posed by seemingly harmless emails and how their innocent actions could lead to their systems being compromised. The Anti Phishing Workgroup (APWG) report states that in the second quarter of 2022, APWG saw a record-breaking 1,097,811 total phishing attacks. The third quarter of 2022 saw 1,270,883 phishing assaults in total, setting a new record and ranking as the worst quarter for phishing ever recorded [1].

Phishing detection models have made significant strides in recent years, contributing to improved cybersecurity. Machine learning techniques, particularly deep learning algorithms, have been at the

forefront of these developments [2,3]. These models are designed to analyze various features of phishing emails or websites to identify fraudulent attempts. However, they continue to grapple with several persistent challenges. The dynamic nature of phishing attacks, where threat actors constantly adapt their tactics and evolve their techniques, poses a substantial hurdle [4]. Additionally, the sheer volume of data generated by email and online communications necessitates the development of efficient models capable of processing and analyzing this data in real-time [2]. Furthermore, the need for accurate labeling of training data and the presence of highly sophisticated phishing campaigns makes it challenging to build models with consistently high detection rates [5].

Overcoming these challenges remains a top priority for researchers and practitioners in the field, as the battle against phishing attacks continues to evolve in complexity. These phishing detection methods frequently have a high proportion of false positives and poor detection accuracy, in particular when faced with fresh phishing tactics [6]. This study contributes to improving the accuracy of phishing detection models by integrating sentence-level features in phishing detection. The focus of the study involves scrutinizing individual sentences within suspicious emails or messages and email header subjects to identify potential signs of phishing. The significance of sentence-level analysis in phishing detection models cannot be overstated. Recent research has shown that traditional approaches focusing solely on analyzing entire emails or web pages may miss subtle cues and indicators of phishing attempts [1,3]. Malicious actors often employ sophisticated techniques, dispersing malicious content across different parts of a message, making it challenging to detect at the email or webpage level alone [7]. By delving into the granular details of sentences within messages, sentence-level analysis enables the identification of specific linguistic and contextual anomalies, increasing the accuracy of phishing detection [8]. Such an approach not only helps in capturing deceptive intent but also enhances the adaptability of detection models to the ever-evolving tactics employed by phishing attackers, ultimately strengthening the resilience of cybersecurity systems against this pervasive threat. By scrutinizing the intricate linguistic and contextual cues inherent to sentences, this study aimed to uncover the subtle linguistic nuances that betray phishing attempts. Integrating machine learning techniques further enhances the efficacy of detection, promoting adaptive cybersecurity measures against the evolving threat landscape.

The paper is organized into five sections. Section 1-Introduction, Section 2-Literature review, Section 3-Methodology, Section 4-Discussion and results and Section 5-Conclusion and future work.

2 Literature Review

2.1 Definition of Phishing

Phishing emails are a specific subset of spam messages when a perpetrator, also known as an attacker, sends victims bogus emails purporting to be from reliable companies. The purpose of these emails is to infect users' systems with malware by concealing malicious attachments or destructive Uniform Resource Locators (URLs) [9].

Phishing constitutes a form of social engineering intrusion, whereby attackers exploit technical means like email and websites to trick visitors into providing their personal data. Phishing emails encompass a blend of social manipulation and technical deception. Phishing emails strive to persuade computer users into sharing sensitive data, such as credit card numbers, login credentials, and other pertinent details. Typically dispatched to numerous randomly chosen recipients, phishing emails tap into users' innate emotions, including envy, fear, respect for authority, curiosity, sympathy, and willingness to engage.

2.2 Phishing Detection Techniques

Phishing attacks are a prevalent form of cybercrime where malicious actors attempt to trick individuals into revealing sensitive information, such as usernames, passwords, or financial data. Various algorithms and techniques have been employed to detect and mitigate phishing attacks.

i) **Heuristic-Based Approaches:** These approaches use rule-based heuristics to identify phishing emails. They often look for common phishing indicators such as misspelled URLs, suspicious sender addresses, and mismatched URLs in email content.
ii) **Blacklist-Based Filters:** Phishing websites and email domains are often added to blacklists. Email and web filters can check incoming data against these lists to block known phishing sources.
iii) **Machine Learning Algorithms:** Machine learning techniques have been widely used for phishing detection. Algorithms like Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks have been applied to classify emails or URLs as phishing or legitimate based on various features.
iv) **Text Analysis:** Natural language processing (NLP) techniques can analyze the text content of emails or webpages to identify suspicious language patterns, keywords, or content structures commonly associated with phishing.
v) **URL Analysis:** Algorithms can analyze URLs to check for deceptive domain names, subdomains, and redirects. Features such as URL length, domain age, and the use of non-standard characters are considered.
vi) **Behavioral Analysis:** Some advanced systems monitor user behavior and detect anomalies. For example, they can identify if a user is suddenly providing sensitive information on a webpage that they have not visited before.

2.3 KNN in Sentence Level Analysis for Phishing Detection

Sentence-level analysis in phishing detection using K-Nearest Neighbors (KNN) involves classifying individual sentences within emails or text data as either indicative of phishing (malicious) or not. KNN, leverages the model's ability to capture patterns and similarities within sentences to identify potentially malicious content. The effectiveness of this method depends on the quality of features, the choice of 'K,' and the distance metric, which may require fine-tuning to achieve optimal results.

2.4 Sentence Level Features in Phishing Detection

Sentence-level features encompass a range of linguistic, structural, and contextual attributes within individual sentences. This approach recognizes that phishing emails often contain subtle cues indicative of malicious intent that can be missed when examining the email as a whole. Sentence-level analysis plays a crucial role in phishing detection by examining individual sentences within email messages or other forms of communication to identify signs of phishing attempts. This analysis involves scrutinizing various linguistic cues and contextual elements to determine whether a sentence is likely to be part of a phishing scheme.

i) Linguistic Cues and Stylistic Anomalies: Phishing emails frequently exhibit linguistic inconsistencies, such as misspellings, grammar errors, and unnatural language usage. Studies by [10] and [11] have explored the linguistic features that are indicative of phishing. These features include grammatical errors, unusual syntax, and inconsistent language usage, which are often present in phishing emails. Sentence-level analysis identifies and quantifies these anomalies, enabling algorithms to accurately differentiate between genuine communication and phishing attempts [12].

Table 1 below outlines linguistic cues used in the study. It highlights the main cues selected in extraction of email content. It entails the linguistic cue, its definition and an example extracted from the dataset.

Table 1: Linguistic cues

Linguistic cue	Definition	Example
Misspellings and grammatical errors	Misspelled words and grammatical errors that can be a red flag	• "Please verify your account detalis." • "Click here to avoid your account being suspensed."
Unnatural language usage	Language that is unnatural or overly formal, attempting to appear official	• "Dear Sir/Madam, we hereby request your immediate action." • "We have noticed some unusual activities on your account, hence we are taking preventive measures."
Urgent or alarmist language	Sense of urgency or panic to prompt immediate action	• "Your account has been compromised. Act now to prevent further damage!" • "Immediate action required: Your account will be suspended in 24 h."
Unusual requests	Unusual requests or demands that are not typically seen in legitimate communications	• "Please provide your password for verification purposes." • "Transfer funds to this account to claim your prize."
Generic greetings	Use of generic greetings to address recipients	• "Dear Customer" • "User"
Generic content	Lack of specific details about the recipient or transaction	• "Your recent purchase requires confirmation. Click here to verify."
Threats or consequences	Negative consequences if action is not taken	• "Failure to update your information will result in account suspension." • "Your account will be charged unless you confirm your details."

ii) Contextual Indicators: Contextual features involve examining the relationship between sentences within an email. Phishing emails often employ urgency, fear, or a sense of authority to manipulate recipients. By considering the connections between sentences, models can identify patterns that suggest phishing intent [13]. Contextual indicators in sentence-level analysis for phishing detection involve examining the relationship between sentences within an email. This study explored features such as:

- **Urgent Action and Threats:** Contextual cues involving urgent language and threats can signal phishing attempts. Identified sentences include "Your account has been compromised. Act now to prevent unauthorized access." [13] and "Failure to respond within 24 h will result in account suspension." [14].
- **Request for Sensitive Information:** Contextual cues where preceding sentences ask for personal information can be suspicious: "To verify your account, we need your username and password. Please provide them." [15] and "For security purposes, we require your Social Security number. Kindly share it." [12].

- **Promised Rewards or Benefits:** Contextual cues that promise rewards can be indicative of phishing attempts: "Congratulations! You've won a gift card. Click the link to claim your prize." [13] and "As a valued customer, you've been selected for a special offer. Click here to redeem." [15].
- **False Authority or Impersonation:** Contextual cues where the email claims authority or impersonates an official source: "This email is from the IT department. Your password needs to be updated." [14] and "We've detected fraudulent activity. Please follow these steps to secure your account." [12].
- **Link and Attachment Context:** Contextual cues involving hyperlinks or attachments can be analyzed to identify phishing: "Click the link below to verify your account information." [15] and "Open the attached file for important account updates." [12].
- **Inconsistent Details:** Contextual cues where information contradicts earlier statements can be suspicious: "You've won a prize in our lottery. However, we need a deposit to process it." [12] and "Your account has been compromised, but we can fix it if you provide your details." [15].

2.5 Phishing Detection Models in Literature

Phishing threats persist as a persistent challenge in cybersecurity, prompting the use of machine learning techniques to identify deceptive activities. The K-Nearest Neighbors (KNN) algorithm emerges as a noteworthy approach applied to phishing detection. This literature review provides an encompassing synopsis of studies employing the KNN algorithm for this purpose.

Sheetal et al. delved into diverse machine learning techniques, encompassing KNN, to discern phishing websites. Their work underscores KNN's role in refining the precision of phishing detection through feature selection and classification [16].

Sharma et al. specifically concentrated on utilizing KNN to identify phishing websites. They elaborated on incorporating URL and webpage content features into the KNN algorithm to achieve effective classification [17].

Anwar et al. proposed an elevated phishing detection model, merging feature selection techniques with KNN. Their research expounds on refining KNN's performance by selecting pertinent features, thereby heightening detection accuracy [18].

Shailendra et al. navigated the realm of identifying phishing webpages through KNN. They underscored the significance of adept feature selection and extraction in bolstering the algorithm's effectiveness [19].

Priyank et al. examined the utility of the KNN algorithm in detecting phishing attacks. Their study emphasized the pivotal role of feature engineering in selecting relevant attributes for classification purposes [20].

The studies collectively illuminate the significance of KNN as a tool for detecting phishing activities, showcasing the ways in which feature selection, engineering, and classification contribute to the advancement of this cybersecurity endeavor.

3 Methodology

3.1 Study Objective

The objective of this research was to develop a sentence level analysis model for detecting phishing in email messages. The research study focused on examining sentence level features at the header of the emails in the dataset and the body of the email. The study culminated in designing and validating

a phishing detection model based on the sentence level features and KNN classifier. The performance of the built model was then evaluated using accuracy, precision, recall, F1-score and ROC Curves. Experimental research design was adopted for the study. The dataset was extracted from Kaggle an open access dataset repository consisting of a large collection of spam and ham email messages which was ideal for this study. The dataset is a csv file with associated data from 5171 randomly selected email files and their corresponding labels for classification as spam or not-spam. Each row in the 5171-row csv file represents a single email. To safeguard privacy, the name has been specified with numbers rather than the receivers' names.

3.2 Experimental Steps

This study focused on extracting sentence level features from email subjects in the header and email body content to realize a better phishing detection model solution. For sentence-level analysis, the study used Gensim library in Python to train a Word2Vec model on a large corpus of text data. This model generated word embedding that captured semantic relationships between words. The trained Word2Vec model was fitted into the KNN classifier whose purpose was to classify a text as spam or ham depending on the message content. Fig. 1 below represents a flow chart outlining the overall methodological steps adopted in the study. The steps are as follows: Loading the datasets into the model. Data preprocessing which entailed tokenization-stop-words removal, lowercasing and vectorization. Data splitting into training set and testing set: 80:20. With KNN classifier the email message is compared to check whether it is a ham or spam and the result is outputted.

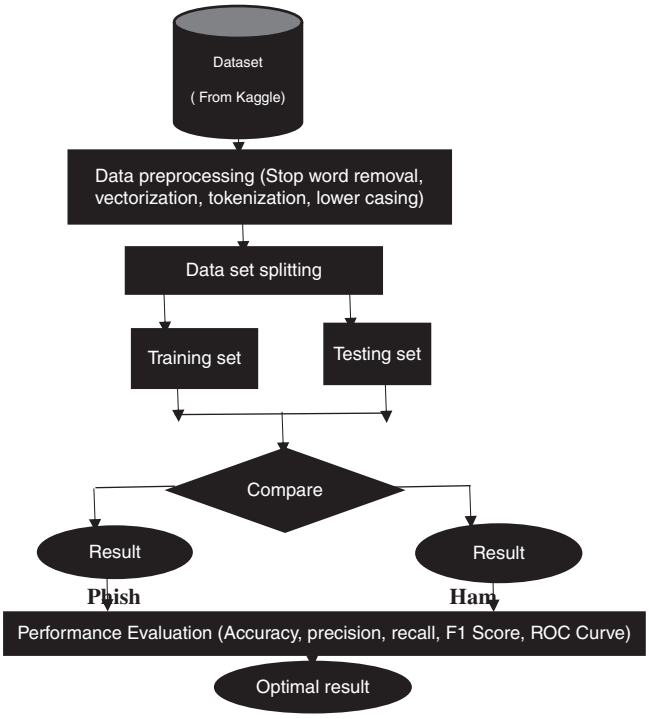

Figure 1: Overall methodology

Fig. 2 represents the proposed model diagram. It starts with email messages that are from the emails subject at the header and email content from the body. The sentence tokens are fed into Word2Vec a vectorization tool and the sentence vector are obtained. The vector code is used to train

the model, and the model is tuned for better results in the KNN classifier. The testing set data is then used to test and validate the model, to check the output whether phish or ham.

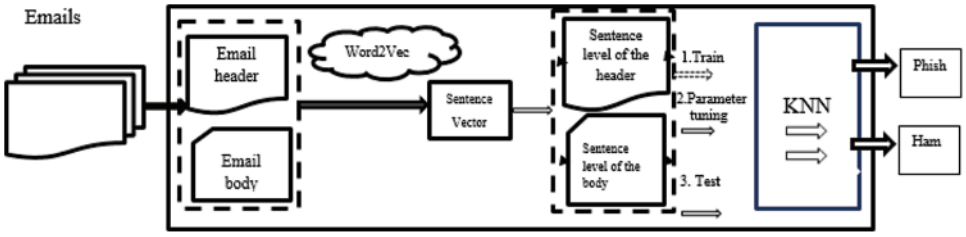

Figure 2: Proposed sentence level analysis model

3.3 Dataset Selection

Dataset used in the study were obtained from open access Kaggle repository (https://www.kaggle.com/datasets/venky73/spam-mails-dataset) [21]. Fig. 3 below shows the raw dataset before processing. The raw dataset from Kaggle is contained in a csv file. It outlines the email message label either ham or spam. The subject of the email and the code for either ham coded 0 and spam coded as 1.

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

Figure 3: Snippet of raw dataset

3.4 Data Preprocessing

This phase entailed tokenization, which entails breaking the sentences up into words and changing a word's case to lower case. Removal of stop words, which are frequently used words (such as a, an, the, etc.) in documents, and use of contractions that were used to expand English contractions to their original form. These words do not actually mean anything because they do not aid in the separation of two papers or the removal of html tags. Email communications that had been processed and categorized as spam (denoted by 1) and ham (denoted by 0) were the results of data preprocessing.

3.4.1 Splitting the Dataset into Training and Testing Set

The preprocessed dataset was split into training and testing set, such that 20% was used for testing and the remaining 80% was used for training. The training set was used for training the model while the testing set was to evaluate the model performance.

3.4.2 Feature Extraction Using Word2Vec

We employed the Gensim library in Python to train a Word2Vec model on a large corpus of text data. This model generated word embedding that captured semantic relationships between words. For

each email subject and sentence in the email body, the average Word2Vec vector for all words was calculated resulting in a sentence-level feature vector. Word2Vec is a Natural Language Processing model, that provides a numerical vector representation for a given word. This numerical vector is often called as "Word Embedding". For each text example in the dataset, we used the trained Word2Vec model to obtain vector representations of the text. We then averaged the Word2Vec word vectors for each word in a sentence to get a sentence-level vector representation. The word embedding procedure is depicted in Fig. 4 below. Only textual content was considered at this level, email addresses and URL were not used in the study.

```
from gensim.models import Word2Vec

words_in_sentences=[]
for i in tqdm(x_train):
    words_in_sentences.append(i.split())

100%|██████████| 4136/4136 [00:00<00:00, 175907.45it/s]

model = Word2Vec(sentences=words_in_sentences, vector_size=200,workers=1, min_count=4)

model.wv.most_similar('lottery', topn=10)

[('foot', 0.9934641122817993),
 ('servers', 0.9929482936859131),
 ('targeted', 0.9928004741668701),
 ('vision', 0.9923527240753174),
 ('fund', 0.9922614693641663),
 ('screens', 0.9915757179260254),
 ('rights', 0.9915533065795898),
 ('hospitals', 0.9912154078483582),
 ('benefit', 0.9912105202674866),
 ('happenstance', 0.9911220669746399)]
```

Figure 4: Word2Vec model

3.5 Training of the Model

This process involved training the model using KNN classifier by imputing the vectorized training data and the labels for spam/ham. KNN model was developed and fitted with the trained Word2Vec model vectors for spam and ham. KNN algorithm places the new case in the category that is most similar to the available categories based on the assumption that the new instance and the data are comparable to the examples that are already accessible. The classifier saves all the information that is available and categorizes new input based on similarity. This means that as fresh data is generated, then quickly categorized into a suitable category using the KNN method. Since KNN is a non-parametric technique, it makes no assumptions about the underlying data. The KNN method simply retains the dataset during the training phase, and when it receives new data, it categorizes it into a category that is very comparable to the incoming data. The accuracy of the test and the fitting of the KNN algorithm to the training set are depicted in Fig. 5 below. "scikit-learn library" was imported to implement the KNN classifier algorithm. The classifier is then trained using the sentence-level Word2Vec representations. The **k** parameter was adjusted, which represents the number of neighbors to consider when making predictions. The experiment was conducted with different values of **k** to find the one that works best

for the dataset. After fitting the KNN classifier on the training data, we used it to make predictions on the test data.

```
from sklearn.model_selection import RandomizedSearchCV
from sklearn.neighbors import KNeighborsClassifier
grid_params = { 'n_neighbors' : [10,20,30,40,50,60],
                'metric' : ['manhattan']}
knn=KNeighborsClassifier()
clf = RandomizedSearchCV(knn, grid_params, random_state=0,n_jobs=-1,verbose=1)
clf.fit(x_train_transformed,y_train)
```

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:305: UserWarnin
  warnings.warn(
▸       RandomizedSearchCV
▸ estimator: KNeighborsClassifier
```

```
clf.best_params_
```

```
{'n_neighbors': 10, 'metric': 'manhattan'}
```

```
clf.best_score_
```

```
0.9470503361781424
```

Figure 5: KNN classifier code snippet

The study utilized the KNN algorithm with a specific value for the "k" parameter. Additionally, we employed the Manhattan distance metric as a measure of similarity between sentences. This combination of parameter settings allowed us to achieve a robust and effective approach for identifying phishing attempts by analyzing sentences within emails.

3.6 Model Evaluation

A prototype was developed to demonstrate how the proposed model works. Python programming language was used since it is a simple language and works well with enormous programs. We evaluated the performance of our approach using a confusion matrix. The confusion matrix allowed us to calculate various metrics such as accuracy, precision, recall, F1-score, and specificity. The study adopted the following definitions:

- **True Positive (TP):** Phishing sentence correctly classified as phishing.
- **True Negative (TN):** Legitimate sentence correctly classified as legitimate.
- **False Positive (FP):** Legitimate sentence incorrectly classified as phishing.
- **False Negative (FN):** Phishing sentence incorrectly classified as legitimate.

The model's performance was further evaluated for accuracy using custom data. Fig. 6 below depicts that the model is able to identify a spam message. The trained KNN classifier was used to predict whether an email is spam or not spam. After training the KNN classifier, we used it to make predictions on new emails by first extracting features from the new email and then using the **predict** method we classified emails as phishing or legitimate.

```
message=['you have worn 5000, call or email to claim your prize']
x_test_transformed2=avg_w2vec(message)

category = clf.predict(x_test_transformed2)
print("The message is", "spam" if category == 1 else "not spam")

100%|██████████| 1/1 [00:00<00:00, 174.95it/s]The message is spam
```

Figure 6: Model output screenshot

4 Discussion and Results

This section outlines the significant outcomes of the study achieved in the developed model. The model was evaluated using the standard performance metrics; accuracy, precision, recall and F1-score. Accuracy evaluation was done by splitting the training and testing dataset in the ratio 80:20. Fig. 7 below shows the results of the model evaluation. The testing accuracy of 97% is ideal for the model. The choice of the "Manhattan distance" as the distance metric contributed to the good model's accuracy in detecting deceptive elements within sentences, further enhancing the security posture of organizations and guarding against attack. Standard evaluation metrics along with the percentages of phishing and non-phishing samples, helped us to assess the performance of our phishing detection model at the sentence level and understand the distribution of samples in the test set.

```
[ ] print(classification_report(y_train,clf.predict(x_train_transformed)))
```

	precision	recall	f1-score	support
0	0.96	0.97	0.96	2460
1	0.92	0.90	0.91	1004
accuracy			0.95	3464
macro avg	0.94	0.93	0.94	3464
weighted avg	0.95	0.95	0.95	3464

Figure 7: Evaluation metrics

The model presented testing accuracy of 97% and a training accuracy of 99% as shown in Fig. 8. The ROC curve helped us to evaluate the model's ability to distinguish between phishing and non-phishing sentences at different probability thresholds. We calculated and printed the accuracy of the model, which represented the overall correctness of the predictions. The ROC curve and accuracy were important metrics for assessing the performance of the KNN classifier for phishing detection at the sentence level.

In addition, the confusion matrix provided insights into the model's performance across different metrics. The model attained a 97% accuracy which was found satisfactory as earlier indicated. The accuracy, precision, recall, and F1-score were computed based on the values from the confusion matrix. Table 2 is a representation of the confusion matrix for our KNN classifier, which provided insights into the model's performance in terms of true positives, true negatives, false positives, and false negatives for both phishing and non-phishing sentences.

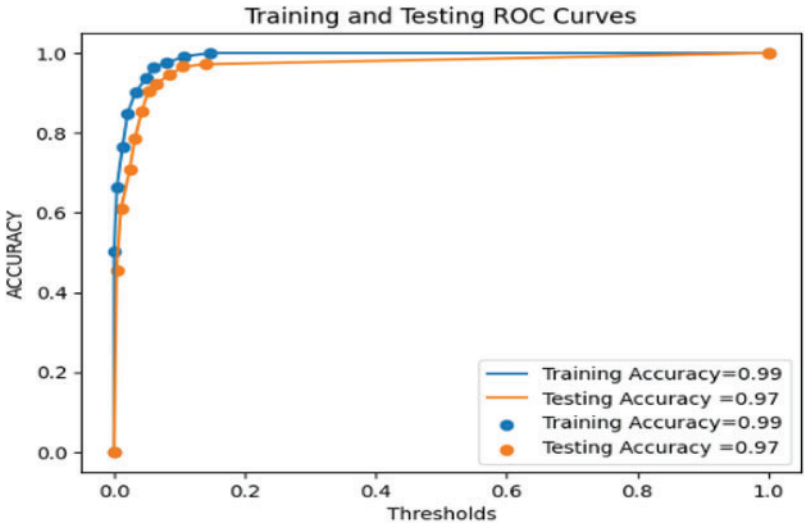

Figure 8: Training and testing ROC curve

Table 2: Confusion matrix

	Predicted legitimate	Predicted phishing
Actual legitimate	2800	97
Actual phishing	89	1100

The overall findings show that the model achieved a 97% accuracy rate using sentence-level features for phishing detection through the use of a K-Nearest Neighbors (KNN) classifier. This robust performance suggests that the model is proficient in distinguishing between legitimate and phishing sentences extracted from emails. Table 3 below compares the model's accuracy percentage with other models in literature. From the comparative analysis it is evident that the integration of sentence-level features with the K-Nearest Neighbors (KNN) algorithm in phishing detection represents a significant advancement. By incorporating sentence-level features, the model gains a deeper understanding of contextual nuances and linguistic patterns, enabling it to identify even subtle phishing attempts. This flexibility, combined with the algorithm's adaptability to various sentence structures extends the applicability of the model in emails, particularly when dealing with short and context-dependent email content.

Table 3: Comparative accuracy analysis of phishing detection models in literature

Anti-phishing method	Authors	Techniques	Dataset source	Accuracy
Context-based	Lindah et al.	Sentence level features, word embedding, KNN	Kaggle	97%

(Continued)

Table 3 (continued)

Anti-phishing method	Authors	Techniques	Dataset source	Accuracy
Content-based	Jain et al. [22]	Modified TF-IDF	Alexa dataset, Open Phish, Phish Tank	89%
	Sonowal et al. [23]	PhiDma framework incorporates five layers	Phishload, 2016. Legitimate URL dataset	92.72%
Machine learning	Chiew et al. [24]	Cumulative Distribution Function gradient (CDF-g), Random Forest, SVM, Naive Bayes, C4.5, JRip, and PART	UCI phishing datasets	94.60%
	Sahingoz et al. [25]	Random forest with NLP	PhishTank, Yandex	97.98%
Deep learning	Adebowale et al. [26]	CNN and LSTM	PhishTank, Common crawl	93.28%
	Wang et al. [27]	LSTM and CNN	Alexa, PhishTank	97%
Fuzzy recognition	Zabihimayvan et al. [28]	Fuzzy Rough Set (FRS)	UCI1, UCI2, Mendeley	95% Fmeasure
	Pham et al. [29]	Neuro-fuzzy, Fog computing, Cloud computing	PhishTank, DMOZ	98.36%
Hybrid learning	Ali et al. [30]	Deep neural networks (DNNs) and genetic algorithm (GA)	UCI phishing websites	91.13
	Zhu et al. [31]	Decision Tree and Optimal Features based Artificial Neural Network, K-medoids clustering algorithm	UCI, PhishTank, Alexa	95.76%
Data mining	Subasi et al. [32]	Random Forest	UCI, WEKA	97.36%
	Sentürk et al. [33]	Decision tree with J48 algorithm	WEKA	89%
	Feng et al. [34]	Monte Carlo algorithm	UCI	97.71%

5 Conclusion and Future Work

The Sentence Level Analysis model's ability to identify nuanced patterns within sentences, coupled with the KNN algorithm's simplicity and interpretability, contributes to its potential as a robust phishing detection tool. The model's focus on sentences provides a fine-grained examination of linguistic cues and context, further strengthening its performance. The study has made noteworthy

contributions to the domain of phishing detection through sentence-level analysis employing the KNN algorithm. Our research has shown that KNN can significantly improve detection accuracy compared to other models, providing a more granular understanding of deceptive elements within phishing emails and websites. Theoretical implications of this study include advancing our understanding of phishing techniques and highlighting the effectiveness of KNN in detecting them, paving the way for further exploration of machine learning and natural language processing techniques in cyber threat detection. From a managerial perspective, our findings offer valuable insights for organizations, email users and cybersecurity professionals, emphasizing the importance of incorporating sentence-level analysis with KNN into anti-phishing strategies to bolster security measures against evolving threats.

However, the model does encounter challenges and limitations. Variability in sentence structure, language nuances, and evolving phishing tactics pose potential hurdles. Additionally, model performance could be influenced by the quality and quantity of training data and varying the K parameter in the classifier necessitating thorough evaluation and fine-tuning.

Future research can explore advanced techniques for generating sentence embedding, utilizing methods like transformer-based models to capture intricate linguistic features and contextual information more effectively. Integrating multiple data modalities, such as images and URLs, along with textual content, could enhance the model's ability to detect phishing attacks across diverse mediums. Other work could focus on investigating adversarial attacks against the Sentence Level Analysis model which can uncover vulnerabilities and foster the development of mitigation strategies for improved resilience.

Acknowledgement: We would like to express our sincere gratitude to all those who contributed to the successful completion of this research paper, "Sentence Level Analysis for Phishing Detection using KNN." We are thankful to the research team members, Lindah Sawe, Joyce Gikandi, John Kamau and David Njuguna, for their collaboration and valuable insights throughout the research process. Our heartfelt appreciation goes to all the individuals and institutions that reviewed and provided constructive feedback on this research paper. Your valuable input helped improve the quality and rigor of our work. Without the collective efforts and support of all these individuals and organizations, this research paper would not have been possible. Thank you all for your invaluable contributions.

Funding Statement: No specific grant from a funding organization supported this research.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Lindah Sawe; training and testing of the algorithm: David Njuguna and Lindah Sawe; analysis and interpretation of results: Lindah Sawe, Joyce Gikandi, John Kamau and David Njuguna; draft manuscript preparation: Lindah Sawe, Joyce Gikandi, John Kamau and David Njuguna. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this research paper "Sentence Level Analysis for Phishing Detection Using KNN," is obtained from the Kaggle and is publicly available for research purposes. The dataset can be accessed at the following URL: https://www.kaggle.com/datasets/venky73/spam-mails-dataset.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Anti-Phishing Working Group, "APWG phishing activity trends report," 2022. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf (accessed on 23/08/2023).

[2] M. N. Alenezi, H. Alabdulrazzaq, A. A. Alshaher and M. M. Alkharang, "Evolution of malware threats and techniques: A review," *International Journal of Communication Networks and Information Security*, vol. 12, no. 3, pp. 326–337, 2022.

[3] Y. Chen and Y. Yang, "An advanced deep attention collaborative mechanism for secure educational email services," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, 2022.

[4] H. B. Abdul Wahab and T. M. Abed, "Detect and prevent phishing based on hybrid approach," *Al-Mansour Journal*, vol. 33, pp. 1–17, 2022.

[5] L. Chen, J. Peng, J. Liu, J. Li, F. Xie, "Phishing scams detection in ethereum transaction network," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–16, 2020.

[6] A. Philip and B. Kabaso, "Hybrid machine learning: A tool to detect phishing attacks in communication networks," *ECTI Transactions on CIT*, vol. 15, no. 3, pp. 374–389, 2021.

[7] S. Paliath, M. A. Qbeitah and M. Aldwairi, "PhishOut: Effective phishing detection using selected features," in *Int. Conf. on Telecommunications*, Bali, Indonesia, pp. 1–5, 2020.

[8] N. Rifat, M. Ahsan, M. Chowdhury and R. Gomes, "BERT against social engineering attack: Phishing text detection," in *IEEE Int. Conf. on Electro Information Technology*, Mankato, MN, USA, pp. 1–6, 2022.

[9] R. Dhruv and M. Suman, "Detection of E-mail phishing attacks—using machine learning and deep learning," *International Journal of Computer Applications*, vol. 183, no. 47, pp. 1–7, 2022.

[10] J. Yao, C. Wang, C. Hu and X. Huang, "Chinese spam detection using a hybrid BiGRU-CNN network with joint textual and phonetic embedding," *Electronics*, vol. 11, no. 15, pp. 2418–2430, 2022.

[11] D. J. Liu, G. G. Geng and X. C. Zhang, "Multi-scale semantic deep fusion models for phishing website detection," *Expert Systems with Applications*, vol. 209, pp. 1–13, 2022.

[12] D. Gibert, C. Mateu and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, vol. 153, pp. 102526–102539, 2020.

[13] A. Almomani, M. Alauthman, M. T. Shatnawi, M. Alweshah, A. Alrosan, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–24, 2022.

[14] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor and J. Downs, "Who falls for phishing scams? A demographic analysis of phishing susceptibility and effectiveness of interventions," in *Conf. on Human Factors in Computing Systems*, New York, NY, USA, pp. 373–382, 2010.

[15] M. Alazab, R. Abu Khurma, A. Awajan and M. Wedyan, "Digital forensics classification based on a hybrid neural network and the salp swarm algorithm," *Electronics*, vol. 11, no. 12, pp. 1903–1923, 2022.

[16] S. Mehta, V. Kanhangad and T. M. Ravi, "Phishing detection using machine learning techniques," *International Journal of Computer Applications*, vol. 117, no. 22, pp. 9–13, 2015.

[17] D. Sharma and V. Gomase, "An approach for detecting phishing websites based on K-nearest neighbors algorithm," *International Journal of Computer Applications*, vol. 176, no. 3, pp. 22–26, 2017.

[18] M. Anwar, A. K. Sangaiah and M. S. Farooq, "An improved phishing detection model using feature selection and K-nearest neighbor," *International Journal of Information Management*, vol. 40, pp. 76–88, 2018.

[19] S. S. Parihar, J. P. Gupta and V. Kumar, "Phishing detection based on the features of phishing webpages using K-nearest neighbor algorithm," *International Journal of Computer Applications*, vol. 182, no. 2, pp. 38–43, 2019.

[20] P. R. Chaudhari, R. V. Jhaveri and K. V. Maheta, "K-nearest neighbor algorithm for phishing detection," *Procedia Computer Science*, vol. 165, pp. 272–279, 2019.

[21] V. Garnepudi, "Spam mails dataset," 2019. [Online]. Available: https://www.kaggle.com/datasets/venky73/spam-mails-dataset (accessed on 25/06/2023).

[22] A. K. Jain, S. Parashar, P. Katare and I. Sharma, "PhishSKaPe: A content-based approach to escape phishing attacks," in *Proc. of Computing and Network Communications (CoCoNet'19)*, vol. 171, pp. 1102–1109, 2020.
[23] G. Sonowal and K. S. Kuppusamy, "PhiDMA—A phishing detection model with multi-filter approach," *Journal of King Saud University—Computer and Information Sciences*, vol. 32, no. 1, pp. 99–112, 2020.
[24] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. Yong and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, 2019.
[25] O. K. Sahingoz, E. Buber, O. Demir and B. Diri, "Machine learning-based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
[26] M. A. Adebowale, K. T. Lwin and M. A. Hossain, "Deep learning with convolutional neural network and long short-term memory for phishing detection," in *Int. Conf. on Software, Knowledge, Information Management and Applications (SKIMA)*, Island of Ulkulhas, Maldives, pp. 1–8, 2019.
[27] W. Wang, F. Zhang, X. Luo and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 1–15, 2019.
[28] M. Zabihimayvan and D. Doran, "Fuzzy rough set feature selection to enhance phishing attack detection," in *Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA, vol. 2019, pp. 1–6, 2019.
[29] C. Pham, L. A. T. Nguyen, N. H. Tran, E. N. Huh and C. S. Hong, "Phishing-aware: A neuro-fuzzy approach for anti-phishing on fog networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1076–1089, 2018.
[30] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Information Security*, vol. 13, no. 6, pp. 659–669, 2019.
[31] E. Zhu, Y. Ju, Z. Chen, F. Liu, X. Fang *et al.,* "An artificial neural network phishing detection model based on decision tree and optimal features," *Applied Soft Computing*, vol. 95, pp. 106505–106517, 2020.
[32] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," in *Int. Conf. on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, United Arab Emirates, pp. 1–5, 2017.
[33] Ş. Şentürk, E. Yerli and İ. Soğukpınar, "Email phishing detection and prevention by using data mining techniques," in *Int. Conf. on Computer Science and Engineering (UBMK)*, Antalya, Turkey, pp. 707–712, 2017.
[34] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han *et al.,* "The application of a novel neural network in the detection of phishing websites," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2018. https://doi.org/10.1007/s12652-018-0786-3