Detecting Malicious Emails: Machine Learning Techniques Unleashed

Nisar Ali[*], Mansoor Qadir[**], Sadeeq Jan[*], [***]Muhammad Waqas, Ghassan Husnain[**], Muhammad Assam[****], Maimoona Asad[*****]

[*]University of Engineering & Technology, Peshawar, Pakistan
[**]CECOS University of IT & Emerging Sciences, Peshawar, Pakistan
[***]IQRA National University, Peshawar, Pakistan.
[****]Department of Software Engineering, University of Science and Technology Bannu, KP, Pakistan
[*****]Department of Information and Communication Engineering from College of Electronic and Information Engineering, Shenzhen University, China

Abstract- This research investigates novel methods for effectively detecting malicious emails using machine learning techniques. Recognizing the evolving landscape of cyber threats, the study delves into advanced methodologies, employing sophisticated algorithms to discern subtle patterns indicative of malicious intent within email content. The research emphasizes a unique combination of feature engineering and leveraging diverse attributes. Through a comprehensive analysis of a diverse dataset, the study showcases the superior efficacy of the proposed machine learning model in accurately identifying and mitigating the risks associated with malicious emails. The findings contribute to the ongoing discourse on cybersecurity, presenting novel insights and advancements in safeguarding digital communication channels against evolving cyber threats.

Our research endeavors involve the exploration of an effective model for identifying malicious emails, utilizing eight diverse datasets across various dimensions. We employ different feature engineering techniques, including term frequency-inverse document frequency (TF-IDF), count vectorization (CV), and a combination of both TF-IDF and EPOCH. Additionally, we integrate the top three models through stacking and rigorously evaluate the outcomes to enhance our understanding of mitigating phishing threats effectively.

Index Terms- Cybersecurity, Data Analysis, Detection, Classification, Machine Learning, Malicious Emails.

I. INTRODUCTION

The use of email has become a fundamental aspect of contemporary living, enabling smooth communication for both personal and professional interactions. However, the extensive adoption of email has turned it into a primary target for cybercriminals aiming to capitalize on its weaknesses. Malicious emails, various activities like phishing attacks, the distribution of malware, and scams involving social engineering present substantial risks to individuals, businesses, and governmental organizations. Unsolicited or promotional messages are dispatched by utilizing email to a group of recipients where the recipients have not given permission to receive such messages [1]. Crafted with deceitful intent, these harmful emails aim to mislead recipients, resulting in sensitive information to be leaked and used to take money out of accounts, database leaks which are then sold on the dark net.

Enhancing the detection of malicious emails by employing innovative deep-learning architectures specifically designed for the comprehensive analysis of entire email content [2]. The proposed framework undergoes a thorough evaluation, demonstrating superior results with an AUC of 0.993. These outcomes outperform state-of-the-art methods for detecting malicious emails, including feature-based machine learning models designed by human experts, with a True Positive Rate (TPR) improvement of 5%.

In Ref. [3], the authors introduce a method with the Natural Language Processing method aimed at detecting phishing emails by extracting keywords from the message body. The main problem with these two methods is that there will be feature loss in the processing of feature extraction. Therefore, machine learning algorithms cannot accurately detect phishing emails.

This document presents a comprehensive analysis of multiple machine learning algorithms utilized for classification of websites which are intended for phishing. Unlike previous work done in this field, our evaluation entails thorough assessment with association of various approaches focused on identifying phishing websites. Significantly, this research introduces an innovative approach by employing three distinct datasets for the training, testing, and validation of multiple classification algorithms, including DT [4], SVM [5], Random Forest [6], Naïve Bayes (NB) [7], K Nearest Neighbor's (KNN) [8], and Artificial Neural Networks (ANN) [9], to distinguish between legitimate and phishing websites. we also employ the well-recognized Principal Component Analysis (PCA) [10] to reduce dimensionality. This approach achieves classification performance that is either equivalent to or exceeds the results obtained by using the complete feature set in the dataset. Moreover, PCA-derived component loadings in the three datasets is explored for the importance of all characteristics.

Conventional rule-based systems for email filtering have been a traditional method for combating harmful emails. While they can handle known threats by using defined rules and databases containing signatures, but newer threats masked in different code are often evaded. Considering that cybercriminals continuously adjust their strategies, there is an urgent requirement to investigate more sophisticated and flexible solutions for efficiently identifying malicious emails.

The main objective of this research is to investigate the use of machine learning techniques to enhance the discovery of emails with code for malicious intent. Specifically, the research aims to:

- Assess the efficacy of diverse machine learning algorithms in identifying various categories of malicious emails.
- Explore feature extraction and engineering methods to capture relevant patterns and characteristics of malicious emails.
- Explore the possibility of applying natural language processing (NLP) techniques to scrutinize email content for indications of malicious intent.
- Compare the performance of machine learning-based approaches with traditional rule-based email filtering methods.

II. LITERATURE REVIEW

In this section, we explore the notable risks presented by cybercriminals utilizing email as a means to distribute malicious code to devices. This activity poses challenges for both individual users and organizations. Identifying and categorizing such harmful emails, which may involve zero-day attacks and targeted phishing such as spear phishing, poses a formidable challenge. This document introduces a solution that integrates deep learning, utilizing data from email headers and bodies, along with dynamic analysis information, as crucial features. The system is subjected to testing using four distinct language email datasets, mimicking real-world scenarios. It attains acceptable accurateness in identifying harmless spam and email with malicious code.

This section presents the review on the topic of investigating use of machine learning for effective detection of malicious emails. Objective of this section is to recognize current work done, approaches, and developments in the realm of cyber security with techniques regarding machine learning, particularly those employed in addressing malicious emails. Through an examination of pertinent literature, this chapter establishes a basis for comprehending the current state-of-the-art and pinpointing areas of research gaps and opportunities for future exploration.

Identifying websites which are known for phishing is essential in the fight against online fraud. Lately, substantial advancements have been achieved in applying machine learning (ML) and data science techniques in diverse fields, including aerospace [11], security at borders [12], medical technologies [13, 14], processing and recognition of speech [15], detection and recognition of objects [16], detection of cybercrime [17]. Similarly, the field of cybersecurity has witnessed numerous technological advancements aimed at automatically detecting phishing attacks. However, this field needs immense improvement. With the emergence of new techniques employed by malicious attackers, phishing incidents are on the rise [18]. To thwart these attacks, multiple strategies have been devised for detection.

In [19], the author has presented an extensive examination of diverse machine learning algorithms, assessing their performance on various datasets. The statistical results suggest that ANN and RF achieve accuracy of more than 97% and exceed other methods. Moreover, [20] a different technique is suggested which combines a Support Vector Machine (SVM) with Genetic Algorithm (GA) to identify phishing emails on Android devices. The suggested method employs feature selection through the Genetic Algorithm (GA) process and utilizes Support Vector Machine (SVM) for classification. The study results demonstrated that this method attained notable accurateness and had surpassed contemporary methods.

Spam detection is extensively studied, with content-based spam identification being an early approach that relied on defining guidelines for recognizing spam messages [21]. Later, the research concentrated on the application of conventional machine learning algorithms such as Naïve Bayes, SVM, Random Forest, Decision trees, etc., necessitating manually crafted characteristics derived from training data [22]. Agarwal [23] also covered techniques for selecting features in classification tasks. Khorsi [24] outlined different statistics-based methodologies employed for filtering spam emails in a review paper. The results indicated that no individual technique was successful in combating spam, attributed with their essential limitations. Moreover, substantial research has been conducted on conventional spam classifiers, as investigated in [25]. In Table 1 the paper's background and its limitations from 2020 to 2023 has been discussed.

A. Detection of emails on IOT platform

In this publication [26], the study highlights the superior performance of supervised learning algorithms, specifically SVM and Naïve Bayes, in spam detection compared to other models. The research offers in-depth insights into these algorithms and suggests future research directions for email spam detection and filtering. The three prominently utilized learning algorithms logistic regression, Naïve Bayes, and support vector machine (SVM) consistently outshine other models across various discussed studies. However, one limitation of the paper is its omission from exploitation into all machine learning classification models and the absence of consideration for deep learning neural networks.

B. Feature-Based Comparison

In the presented study [27], three distinct datasets and seven machine learning models were examined, and results were compared using various algorithms. The findings indicate that employing a multi-feature algorithm with 50 features achieved the highest accuracy, while a reduced feature set of 20 still yielded high accuracy but proved less effective in detecting phishing emails. A limitation of the research is the reliance on a predefined dataset, and the study only evaluates three datasets, asserting that the optimal results are achieved within this limited dataset scope.

C. Systematic-based review of business-type emails

This paper [28] conducts a thorough examination and assessment of contemporary BEC phishing attacks through a systematic review of the literature. The analysis covers articles from journals and conference proceedings released between 2012 and 2022. Employing a specific search strategy, 38 articles were selected from a pool of 950 for detailed scrutiny. The focus of the investigation includes recent models for BEC phishing detection, machine learning algorithms employed in constructing these models, prevalent datasets in model development, and key features employed in the identification of BEC phishing emails. Notably, the study reveals a commonality in the use of DT, SVM, ANN, NB, and Logistic algorithms as the predominant techniques among researchers, and it emphasizes the absence of a specific dataset,

highlighting the exclusive reliance on a systematic review approach.

D. We are improving malicious email detection through deep learning

In our study [29,30], we introduced a comprehensive ensemble framework comprising diverse deep-learning classifiers to enhance the classification of malicious emails, eliminating the requirement for manually generated features. The suggested framework employs the entire email to automatically generate representative features using deep learning detectors. However, the ensemble framework, while advancing email detection accuracy beyond state-of-the-art models, compromises speed compared to a simpler knowledge-based model. Despite existing efficiency disparities, anticipated GPU advancements are expected to facilitate the integration of the proposed framework as an internal email filter within organizations.

For this research, we gathered published papers from diverse academic journals and conferences within the field of Computer Science, with a specific emphasis on cybersecurity. A total of 37 papers were chosen from online platforms through Google Scholar, as outlined in Table 1. Table 2 present the Strings and Keywords utilized for paper searches during the study, and Figure 1 visually represents this information.

Table 1. List of academic databases and number of initial papers selected

S. No	Name	URLs	No. of Papers
1	ResearchGate	www.researchgate.net	10
2	ScienceDirect	www.sciencedirect.com	5
3	IEEE - Access	ieeexplore.ieee.org	12
4	SpringerLink	link.springer.com	8
5	Other Sources	Multiple Sources	37
Total	Source	#	37

Figure 1. Fig of paper investigate

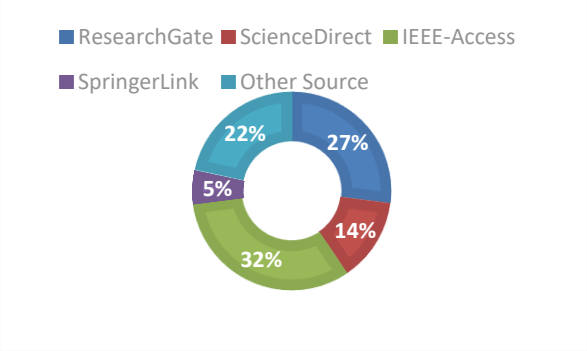


Table 2. Table of Strings/Keywords

S. No	String/Keyword	S. NO	No. of Papers
1	Malicious Email Detection Using Machine Learning	7	Malicious Email Detection Using Ensemble Learning
2	Malicious Email Detection using CNN	8	Voting
3	K-Mean Clustering	9	SVM
4	Random Forest	10	RF
5	Bagging	11	LSTM
6	Ada Boost	12	STACKING

III. METHODOLOGY

A. *Datasets*

To build a comprehensive dataset, this section explains the data collection process, including the sources from which the emails were obtained. It also addresses any legal or ethical considerations involved in data acquisition and ensures the privacy and anonymity of email users. I have collected the datasets from different sources like Kaggle, GitHub, and URLs.

The chapter encompasses details regarding the data sources, spreading of harmful and harmless emails, and size of the dataset. Furthermore, it delves into the criteria for selecting the dataset to ensure its relevance and representativeness in simulating real-world email traffic. The datasets have been collected from various sources such as Kaggle, GitHub, and URLs. These datasets, acquired in different formats with varying characteristics like null values, duplicates, multiple columns, and distinct labels, have undergone a standardization process. They were transformed into a uniform format, eliminating duplicated values and null entries, and retaining only two columns with labels named "test" and "target." Subsequently, the datasets were merged, and any remaining duplicated values were removed to create a consolidated and refined dataset as shown in Table 3.

Table 3. A whole table of all datasets

Datasets	1	2	3	4	5	6	7	8
Dimensions	5572, 2	10239, 2	15405, 2	5293, 2	9687, 2	2591, 2	5293, 2	34055, 2
Size	461kb	12MB	12.9MB	9.98MB	14.2MB	24.4MB	49.7MB	62.6MB
Ham	86.58	57.68	12.9	73.97	50.86	83.67	73.97	63.84
Spam	13.41	42.32	32.37	25.03	49.14	16.33	26.03	36.16

B. *Data Pre-processing*

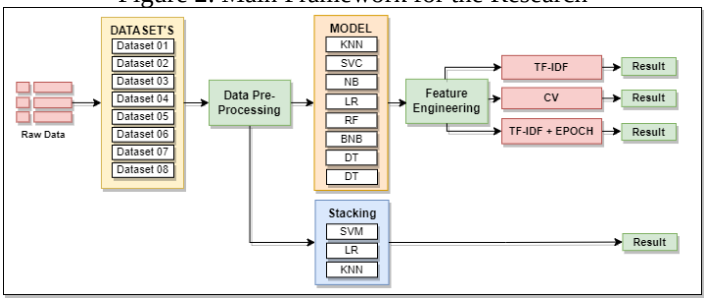
Before feeding the data into machine learning models, data pre-processing is essential to clean and prepare the dataset. This section describes the steps involved in data pre-processing, such as handling missing values, removing duplicates, and addressing any data quality issues. Moreover, it discusses the importance of data balancing techniques to address class imbalance between benign and malicious emails. I have used only one data pre-processing format which is used in TF-IDF, CV, TF-IDF, and EPOCH and I have used NLTK for data pre-processing, word tokenizer for tokenizing the word, stopwords for removing the stopwords, Porter Stemmer use to remove the same word and convert it into only one word. this research is investigating the base; therefore, I have not improved the model by improving or

changing the data pre-processing because when I add something in data pre-processing and then run it on only one model then obviously those models come best.

C. Model Selection

In their upcoming endeavors, the team will delve into the critical phases of model selection, training, and evaluation to construct a formidable email spam classifier. The choice of the most fitting machine learning algorithms from a diverse pool, including K-Nearest Neighbors (KN), Support Vector Classifier (SVC), Naive Bayes (NB), Logistic Regression (LR), Random Forest (RF), Bernoulli Naive Bayes (BNB), Decision Trees (DT), and Gaussian Naive Bayes (GNB), will be meticulously guided by the intricate characteristics inherent within the dataset. Once the ideal model is identified, the team will harness its potential by training it on meticulously preprocessed data, ensuring that it can discern the nuances between spam and legitimate emails with unwavering accuracy. To gauge its performance, the team will employ a suite of essential metrics, including accuracy, precision, recall, and F1 score, thus crafting a robust foundation upon which the email spam classifier can stand. Moreover, they will venture into the realm of hyperparameter tuning, fine-tuning the model's parameters to achieve peak efficiency. Through these strategic undertakings, their objective is not only to streamline email communication but also to fortify security measures, resulting in heightened productivity, enhanced user experiences, and strengthened defenses against the ever-persistent threat of spam emails. In summary, this comprehensive approach, encompassing data preprocessing, feature engineering, model selection, and meticulous evaluation, will culminate in the development of an email spam classifier poised to excel in the real-world email security landscape. Figure 2 illustrates the project's core concept, which involves utilizing eight datasets and taking three models based on feature engineering. The team modified feature engineering in these models, leading to the final statistical outcome. They then examined the most effective machine learning model for identifying malicious emails based on this result.

Figure 2. Main Framework for the Research



D. System Specifications

The testing of all the experiments has been done on a single system mentioned below for the equal evaluation of all the desired targets. They must be tested on a single system so that the memory allocation, power usage, and all other specifications are the same for the investigation for clear and fair analysis. If the specifications are changed, the investigation might result in a different scenario. The results will be changed due to the involvement of processing time, memory allocation, cache memory, GPU usage, memory, operating system, and all other specifications. The main factor is also the use of IDE, which when different IDEs are used on the same system for the analysis might result differently, and also shown in the specs in Table 4.

Table 4. System Specification

System Specification	
Processor	Intel Core i7-4790 CPU @ 3.60GHZ
RAM	16 GB
Hard Disk	256 GB (SSD) + 1 TB (Hard disk)
OS	Windows 10 Pro
GPU	NVIDIA GeForce GT710

IV. Results and Discussion

In this section, the focus is on the discussion and interpretation of the experimental results derived from the investigation into the use of machine learning to detect dangerous emails intended for malicious reasons. The section conducts a thorough analysis of the findings, comparing the performance of various machine learning models and addressing the research objectives and questions.

Figure 53 provides an evaluation of the results based on Accuracy, Precision, Recall, and F1-Score. The authors streamlined the presentation by minimizing the number of tables from 24 to 12. In Table 5, the author counts the frequency of occurrences of each model coming in first place. Random Forest (RF) is noted as the best on precision 10 times, followed by Support Vector Classification (SVC) with 2 occurrences, and K-Nearest Neighbors (KNN) with only 2 instances in the 8 datasets.

Based on Precision, KNN is observed as the top performer, occurring 18 times as the best model, followed by SVM with 17 occurrences, and RF with 7 occurrences in 24 datasets. In terms of Recall, Random Forest is in the lead with 14 occurrences, followed by Decision Tree with 3 occurrences, and SVM with 3 occurrences. Lastly, for F1-Score, Random Forest is again noted as the best model with 10 occurrences, followed by SVM with 3 occurrences, and KNN with 2 occurrences in the 24 datasets. This corresponds to 24 datasets where 3 models are considered multiple times across 4 metrics, resulting in a total of 24 evaluations.

Table 5. Top three model results on A, P, R, and F1 score base

ALGORITHMS	Ist	2nd	3rd
ACCURACY	RF (10)	SVC (2)	KNN (2)
PRECISION	KNN (18)	SVC (17)	RF (7)
RECALL	RF (14)	DT (3)	SVC (3)
F1-SCORE	RF (10)	SVC (3)	KNN (2)

Rank Accuracy	Model	Best in Datasets
1	Random Forest (RF)	5 datasets
2	K-Nearest Neighbors (KN)	1 dataset
3	Support Vector Classifier (SVC)	1 dataset

Rank (Recall)	Model	Best in Datasets
1	Random Forest (RF)	6 datasets
2	Decision Tree (DT)	1 dataset
3	Support Vector Classifier (SVC)	1 dataset

Rank	Model	Best in Datasets
1	Random Forest (RF)	5 datasets
2	Support Vector Classifier (SVC)	1 dataset
3	Logistic Regression (LR)	1 dataset

Rank	Model	Best in Datasets
1	Random Forest (RF)	7 datasets
2	Support Vector Classifier (SVC)	1 dataset
3	Logistic Regression (LR)	1 dataset

Rank	Model	Best in Datasets
1	K-Nearest Neighbors (KN)	1 dataset
2	Decision Tree (DT)	1 dataset
3	Gaussian Naive Bayes (NB)	1 dataset

Rank	Model	Best in Datasets
1	Decision Tree (DT)	2 datasets
2	Random Forest (RF)	2 datasets
3	Support Vector Classifier (SVC)	1 dataset

Rank precision	Model	Best in Datasets
1	K-Nearest Neighbors (KN)	6 datasets
2	Support Vector Classifier (SVC)	2 datasets
3	Gaussian Naive Bayes (NB)	1 dataset

Rank (f1-score)	Model	Best in Datasets
1	Random Forest (RF)	5 datasets
2	K-Nearest Neighbors (KN)	1 dataset
3	Support Vector Classifier (SVC)	1 dataset

Rank	Model	Best in Datasets
1	Support Vector Classifier (SVC)	7 datasets
2	K-Nearest Neighbors (KN)	6 datasets
3	Random Forest (RF)	3 datasets

Rank	Model	Best in Datasets
1	Random Forest (RF)	5 datasets
2	Support Vector Classifier (SVC)	1 dataset
3	Logistic Regression (LR)	1 dataset

Rank	Model	Best in Datasets
1	Support Vector Classifier (SVC)	8 datasets
2	K-Nearest Neighbors (KN)	6 datasets
3	Random Forest (RF)	4 datasets

Rank	Model	Best in Datasets
1	Random Forest (RF)	5 datasets
2	Support Vector Classifier (SVC)	1 dataset
3	K-Nearest Neighbors (KN)	1 dataset

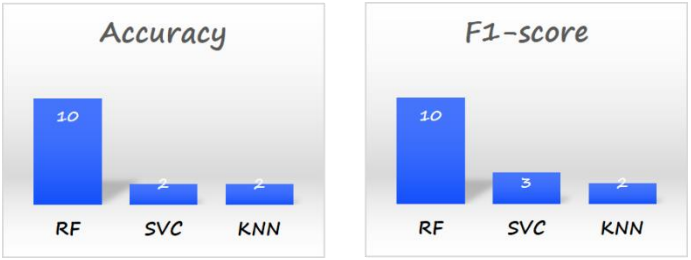
Figure 3. Datasets combined

Table 6 shows that Random Forest (RF) is the most accurate model, followed by Sup-port Vector Classifier (SVC) in the second position and K-Nearest Neighbors (KNN) in the third. This ranking is also reflected in the F1 score. The author does not provide in-depth discussions on precision and recall since the F1-score, represented diagrammatically in Figure 3, already encompasses both metrics, acting as an average of the two.

Table 6. 1st 3 models on Accuracy and F1-Score base

ALGORITHMS	RF	SVC	KNN
ACCURACY	10	2	2
F1-SCORE	10	3	2

Figure 5. Combine table graphs of both Accuracy and F1-Score



.

A. Mathematical Formulation: SVM

For a linearly separable case, the decision function is:

$f(x)=\text{sign}(w\cdot x+b)$ (1)

where:

- w is the weight vector.
- x is the input vector.
- b is a bias term.

The optimization problem to find w and b involves minimizing ‖w‖ (the norm of the weight vector) subject to the constraints that $w \cdot xi+b \geq 1$ for positive examples and $w \cdot xi+b \leq -1$ for negative examples.

The SVM algorithm is a powerful method for classification and regression tasks, particularly effective in high-dimensional spaces. Its ability to handle non-linear relationships through kernel tricks makes it versatile in various machine-learning applications.

B. Mathematical Formulation Random Forest

At each node of the decision tree, an algorithm chooses the feature that maximizes the information gain. The information gained is based on a reduction in entropy, a measure of impurity.

$$Entropy\ (S) = \sum_{i=1}^{c} p_i \log(p_i) \quad (2)$$

where S is a set of examples in a node, c is the number of classes, and is the proportion of examples in the $\sum_j^n \frac{|S_j|}{|S|}$ class i.

Information Gain=Entropy(parent) - * Entropy (S_j), $j = 0,1,2,3.....n$

Gini Impurity (for Classification).

. Another measure of impurity that can be used to split nodes in a decision tree.

$$\text{Gini}\ (S) = 1 - \sum_{i=1}^{c} P_i^2 \quad (1)$$

where S and c have the same meanings as in the entropy formula.

Random Forest Algorithm.
Bootstrapping.
For each tree in the forest, a random sample (with replacement) is drawn from the training data.
Feature Randomization.
To introduce diversity, random subset of features is considered at each node.
Aggregation (Voting or Averaging).
For classification, the final prediction is often determined by a majority vote among the trees. For regression, it's the average of the individual tree predictions.
Notation

- X: Input feature vector.
- Y: Output variable (target).
- N: Number of examples.
- M: Number of features.

Ensemble Prediction.
The prediction of the Random Forest RF(X) is based on the individual predictions $f_i(X)$ from each tree.
Classification

$$RF(X)=\text{mode}\ (f_1(X), f_2(X)..., f_n(X)) \quad (2)$$

Regression

$$RF(X)=\frac{1}{n}\sum_{i=1}^{n} f_i(X) \quad (3)$$

Random Forest is a powerful ensemble method that leverages the diversity of decision trees to improve predictive performance and generalization. The strength of Random Forest lies in its ability to reduce overfitting and provide robust predictions. While the mathematical details of each decision tree are involved, the ensemble nature of Random Forest simplifies the modeling process.

C. Mathematical Formulation K-Nearest Neighbors Algorithm

Notations

- X: Input feature vector.
- Y: Output variable (target).
- N: Number of examples in the training dataset.
- M: Number of features in each example.
- $x^{(i)}$: Feature vector of the i^{th} training example.
- $y^{(i)}$: Target value of the i^{th} training example.
- k: Number of neighbors to consider.

Distance Metrics.
1. Euclidean Distance (for continuous features):

$$d(p,q)=\sqrt{\sum_{i=1}^{M}(p_i-q_i)^2} \quad (6)$$

2. Manhattan Distance (for continuous features):

$$d(p,q)^n=\sum_{i=1}^{M}|p_i-q_i|^2 \quad (7)$$

Hamming Distance (for categorical features):
Count the number of positions at which the corresponding symbols are different.

- *K-Nearest Neighbors Algorithm (Classification)*

1. Find Nearest Neighbors
- For a new data point Xnew, calculate the distance to all training examples.
- $d_i = Distance\ (x_{new}, x(i))$

2. Select k Nearest Neighbors
- Identify k training examples with the smallest distances to X_{new} .

3. Majority Vote

For classification, assign a class label that is the majority among neighbors as the predicted class for X_{new}.

$$\hat{y}_{new}=argmax_y\sum_{i=1}^{k} I(y^{(i)}=y)$$

KNN is a simple, instance-based learning algorithm which predicts outcomes by assessing the resemblance of new data points to pre-existing examples. Its mathematical formulation is largely centered around distance calculations and finding the nearest neighbors.

V. CONCLUSION

This section provides a comprehensive conclusion to the investigation of detecting false and dangerous emails with malicious content with machine learning. It summarizes key findings, contributions, and implications of the research, and discusses the overall significance of the study to the field of email security.

First of all, the individual collected different datasets with varying dimensions, sizes, and data. The datasets were then formatted into a unified structure, featuring two columns where the text and target labels across all datasets were consistently labeled as "text" and "target." There were no additional columns in any dataset, and no instances of duplication or null values were present.

Subsequently, the first and second datasets were merged, resulting in dataset 3, distinguished by its uniqueness due to the absence of duplicated values. Following this, datasets 3, 4, and 5 were merged to create a novel dataset 6, characterized by exclusively unique data achieved through the removal of duplicated values. Ultimately, all datasets were merged to construct an extensive dataset, and various models were employed to determine the most effective one, as outlined in Table 01.

For feature extraction, TF-IDF, Count Vectorization (CV), and TF-IDF Plus EPOCH methods were employed.

Random Forest (RF) consistently emerges as the best model across a diverse range of assessment properties. Notably, RF excels in terms of accuracy, where it consistently achieves remarkable scores, demonstrating its versatility and robustness. Its ensemble approach, combining multiple decision trees to handle various data patterns effectively, solidifies its position as a reliable choice for numerous classification tasks. RF's outstanding performance in capturing complex relationships in the data is particularly noteworthy.

Moreover, when examining precision, the Support Vector Classifier (SVC) emerges as the most effective model. It scores in all datasets reliably and with great precision, which results in less

false and all the while accurate positive. SVC's exceptional precision performance, particularly in cases where precision is paramount, positions it as a top choice for tasks like medical diagnoses or fraud detection.

In terms of recall, the choice of the best model varies depending on the dataset's characteristics. Random Forest (RF), Decision Tree (DT), Support Vector Classifier (SVC), and Gaussian Naive Bayes (GNB) all exhibit their strengths in capturing relevant instances across different scenarios. The selection of the most suitable model should consider the specific requirements of each dataset, as well as the trade-offs between precision and recall.

Lastly, when seeking a balanced performance between precision and recall, the Random Forest (RF) consistently stands out with high F1 scores. RF's ability to strike this balance effectively, combined with its versatility, makes it a compelling choice for tasks that demand overall predictive performance while maintaining a balance between precision and recall.

The choice of the best model should be made by carefully considering the unique characteristics of the dataset, as well as the specific priorities and constraints of the task at hand. While RF consistently shines as a robust performer, it's essential to weigh other factors like model interpretability, computational efficiency, and real-world applicability when making the final decision.

REFERENCES

[1] Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., & Shah, T. (2022). Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges. Security and Communication Networks, 2022, 1-19.

[2] Qabajeh, I., Thabtah, F., & Chiclana, F. (2018). A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Computer Science Review, 29, 44-55.

[3] Muralidharan, T., & Nissim, N. (2023). Improving malicious email detection through novel designated deep-learning architectures utilizing entire email. Neural Networks, 157, 257-279.

[4] Decision Trees — scikit-learn 0.22.1 documentation, Scikit-learn.org. [Online]. Available: https://scikit- learn.org/stable/modules/tree.html. [Accessed: 10- Jun- 2019]

[5] Support Vector Machines — sci-kit-learn 0.22.1 documentation, Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed: 10- Jun- 2019]

[6] Ensemble methods — sci-kit-learn 0.22.1 documentation, Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html

[7] Naive Bayes — scikit-learn 0.22.1 documentation, Scikit-learn.org. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html.

[8] Nearest Neighbors — scikit-learn 0.22.1 documentation, Scikit-learn.org. [Online]. Available: https://scikit- learn.org/stable/modules/neighbors.html

[9] Nicholson, C. (2019). A beginner's guide to neural networks and deep learning. Retrieved January, 30, 2020.

[10] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

[11] Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., & Shah, T. (2022). Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges. Security and Communication Networks, 2022, 1-19.

[12] Mughaid, A., AlZu'bi, S., Hnaif, A., Taamneh, S., Alnajjar, A., & Elsoud, E. A. (2022). An intelligent cyber security phishing detection system using deep learning techniques. Cluster Computing, 25(6), 3819-3828.

[13] Atlam, H. F., & Oluwatimilehin, O. (2022). Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review. Electronics, 12(1), 42.

[14] Muralidharan, T., & Nissim, N. (2023). Improving malicious email detection through novel designated deep-learning architectures utilizing entire email. Neural Networks, 157, 257-279.

[15] Moon, J., Shon, T., Seo, J., Kim, J., & Seo, J. (2004). An approach for spam e-mail detection with support vector machine and n-gram indexing. In Computer and Information Sciences-ISCIS 2004: 19th International Symposium, Kemer-Antalya, Turkey, October 27-29, 2004. Proceedings 19 (pp. 351-362). Springer Berlin Heidelberg.

[16] Khan, W., Ansell, D., Kuru, K., Bilal, M.: 2018. "Flight Guardian: Autonomous Flight Safety Improvement by Monitoring Aircraft Cockpit Instruments", Journal of Aerospace Information Systems, AIAA, 15:203-214

[17] O'Shea, J., Crockett, K., Khan, W., Kindynis, P., Antoniades, A., Boultadakis, G.: 2018. "Intelligent Deception Detection through Machine Based Interviewing", International Joint Conference on Neural Networks (IJCNN)

[18] Khan, W., Badii, A.: 2019. "Pathological Gait Abnormality Detection and Segmentation by Processing the Hip Joints Motion Data to Support Mobile Gait Rehabilitation", Journal of Research in Medical Sciences, 07:1-9

[19] Khan, W., Kuru, K.: 2017. "An Intelligent System for Spoken Term Detection That Uses Belief Combination", IEEE Intelligent Systems, 32:70-79 DOI Publisher Url

[20] Khan, W., Hussain, A., Khan, B., Shamsa, TB., Nawaz, R.: 2019. "Novel Framework for Outdoor Mobility Assistance and Auditory Display for Visually Impaired People", 12th International Conference on the Developments in eSystems Engineering (DeSE2019: Robotics, Sensors, Data Science and Industry 4.0.)

[21] Kuru, K., Khan, W.: 2018. "Novel hybrid object-based non-parametric clustering approach for grouping similar objects in specific visual domains", Applied Soft Computing, Elsevier, 62:667-701 34. Davis, J.: 2019. "Phishing Attacks on the Rise, 25% Increase in Threats Evading Security", HealthITSecurity, [Online]. Available: https://healthitsecurity.com/news/phishing-attacks-on-the-rise-25- increase-in-threats-evading-security

[22] Qadir, H., Khalid, O., Khan, M.U., Khan, A.U., Nawaz, R.: 2018. "An Optimal Ride Sharing Recommendation Framework for Carpooling Services", IEEE Access, 06, 62296-62313, doi: 10.1109/ACCESS.2018.2876595

[23] Ibrahim, D., Hadi, A.: 2017. "Phishing Websites Prediction Using Classification Techniques", International Conference on New Trends in Computing Sciences (ICTCS). DOI: 10.1109/ictcs.2017.38

[24] Mohammad, R. M., McCluskey, T.L., Thabtah, F.: (2012). "UCI Machine Learning Repository", Irvine, CA: the University of California, School of Information and Computer Science. Available: http://archive.ics.uci.edu/ml/datasets/phishing+websites. [Accessed: 16-Jun- 2019] Khan, S.A., Khan, W., Hussain, A. (2020). Phishing Attacks and Websites Classification Using Machine Learning and Multiple Datasets (A Comparative Analysis). In: Huang, DS., Premaratne, P. (eds) Intelligent Computing Methodologies. ICIC 2020. Lecture Notes in Computer Science (), vol 12465. Springer, Cham. https://doi.org/10.1007/978-3-030-60796-8_26

[25] Chen, W., Chen, X., & Yang, X. (2020). A hybrid approach for phishing email detection on Android using GA and SVM. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2991-3002.

[26] Ahmed S, Mithun F (2004) Word stemming to enhance spam filtering. In: The conference on email and anti-spam (CEAS'04) 2004

[27] Agarwal B, Mittal N (2016) Prominent feature extraction for sentiment analysis. Springer International Publishing, Berlin, pp 21–45.

[28] Khorsi A (2007) An overview of content-based spam filtering techniques. Informatica 31(3):269–277.

[29] Kolari P, Java A, Finin T, Oates T, Joshi A (2006) Detecting spam blogs: a machine learning approach. In: Proceedings of the 21st national conference on artificial intelligence (AAAI), July 2006.

[30] Wang AH (2010) Don't follow me: spam detection in Twitter. In: Proceedings of the 2010 international conference on security and cryptography (SECRYPT). IEEE, New York, pp 1–10.

AUTHORS

First Author – Nisar Ali, University of Engineering & Technology, Peshawar, Pakistan.
Second Author – Mansoor Qadir, Ph.D. (CS), CECOS University of IT & Emerging Sciences, Peshawar, Pakistan
Third Author – Sadeeq Jan, Ph.D., University of Engineering & Technology, Peshawar, Pakistan.
Fourth Author – Muhammad Waqas, Ph.D. (EE), Iqra National University, Peshawar, Pakistan
Fifth Author – Ghassan Husnain, Ph.D. (ITS), CECOS University of IT & Emerging Sciences, Peshawar, Pakistan
Sixth Author – Muhammad Assam, Department of Software Engineering, University of Science and Technology Bannu, KP, Pakistan
Seventh Author – Maimoona Asad, MS (Computer Engg), Department of Information and Communication Engineering from College of Electronic and Information Engineering, Shenzhen University, China.

Correspondence Author – Mansoor Qadir