future internet

Article

An Effective Ensemble Approach for Preventing and Detecting Phishing Attacks in Textual Form

Zaher Salah [1,*], Hamza Abu Owida [2], Esraa Abu Elsoud [3], Esraa Alhenawi [3], Suhaila Abuowaida [4] and Nawaf Alshdaifat [5]

[1] Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University, Zarqa 13133, Jordan
[2] Department of Medical Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Ammman 19328, Jordan; h.abuowida@ammanu.edu.jo
[3] Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa 13100, Jordan; eabuelsoud@zu.edu.jo (E.A.E.); ealhenawi@zu.edu.jo (E.A.)
[4] Department of Computer Science, Faculty of Information Technology, Al Al-Bayt University, Mafraq 25113, Jordan; suhila@aabu.edu.jo
[5] Faculty of Information Technology, Applied Science Private University, Amman 11937, Jordan; n_alshdaifat@asu.edu.jo
* Correspondence: zaher@hu.edu.jo

Abstract: Phishing email assaults have been a prevalent cybercriminal tactic for many decades. Various detectors have been suggested over time that rely on textual information. However, to address the growing prevalence of phishing emails, more sophisticated techniques are required to use all aspects of emails to improve the detection capabilities of machine learning classifiers. This paper presents a novel approach to detecting phishing emails. The proposed methodology combines ensemble learning techniques with various variables, such as word frequency, the presence of specific keywords or phrases, and email length, to improve detection accuracy. We provide two approaches for the planned task; The first technique employs ensemble learning soft voting, while the second employs weighted ensemble learning. Both strategies use distinct machine learning algorithms to concurrently process the characteristics, reducing their complexity and enhancing the model's performance. An extensive assessment and analysis are conducted, considering unique criteria designed to minimize biased and inaccurate findings. Our empirical experiments demonstrates that using ensemble learning to merge attributes in the evolution of phishing emails showcases the competitive performance of ensemble learning over other machine learning algorithms. This superiority is underscored by achieving an F1-score of 0.90 in the weighted ensemble method and 0.85 in the soft voting method, showcasing the effectiveness of this approach.

Keywords: phishing email detection; machine learning; ensemble learning; cybersecurity

check for updates

Citation: Salah, Z.; Abu Owida, H.; Abu Elsoud, E.; Alhenawi, E.; Abuowaida, S.; Alshdaifat, N. An Effective Ensemble Approach for Preventing and Detecting Phishing Attacks in Textual Form. *Future Internet* **2024**, *16*, 414. https://doi.org/10.3390/fi16110414

Academic Editor: Francesco Buccafurri

Received: 6 October 2024
Revised: 4 November 2024
Accepted: 6 November 2024
Published: 8 November 2024

1. Introduction

The extensive proliferation of emails poses an increasing and dangerous menace. One of the most harmful tactics used by cybercriminals to carry out destructive activities is phishing attacks via email. This type of attack is considered the first step in launching more extensive and complex attacks, such as Advanced Persistent Threats (APTs) and Exploit Kits [1]. Although phishing email attacks have persisted for a long time, conventional email filtering methods have proven ineffective in successfully countering such attacks. According to recent phishing studies, 75% of organizations experienced a phishing assault in 2020, and 96% of these attacks were carried out using emails [2]. Since COVID-19 occurred in this period, cybercriminals have taken advantage of people's desire to know more about this pandemic to trick them into clicking on harmful websites or files. The phishing attempts associated with COVID-19 were highly successful in tricking victims into thinking they were authentic sources by posing as governments, health centers, ministries of health, or other prominent organizations in a relevant country [2].

Phishing emails have been studied by the academic community for over a decade, during which researchers have suggested several approaches to identify and prevent phishing email attacks, mostly using machine learning techniques. Conversely, recent studies by [3–5] have pointed out several holes as well as drawbacks in the existing literature. These drawbacks comprise a limited range of classifiers, which is particularly pertinent in addressing the imbalanced nature of phishing detection, necessitating extensive utilization of balanced datasets. Additionally, there are inadequate evaluation metrics and dataset-related challenges, such as limited data sources for research. The drawbacks above hinder the practicality of the current works in real-world situations. Additionally, to extract important features, earlier machine learning models focused mostly on either the email's structural properties (headers, URLs, grammar, and attachments) or its body text (text analysis and natural language processing).

Phishing detection is now known as a crucial area of cybersecurity, and many industry-standard products, like Microsoft Defender, Google Safe Browsing, and PhishTank, offer fundamental protections against these attacks. These tools often use behavior analysis, signature-based techniques, and blacklist databases to detect and stop phishing attempts. However, these methods frequently fail to identify novel or subtle phishing variations that evade conventional filters. In response to this challenge, we suggest using ensemble learning (EL) which is considered one of the most popular approaches used for handling these drawbacks. The application of EL involves the integration of multiple machine learning algorithms, and has shown significant advancements in various research domains, including detecting phishing web pages [6] and fraud detection. Table 1 presents the evolution of phishing attacks over the years. This study is motivated by the fact that phishing attacks are becoming more sophisticated, making many of the detection methods that are now in use obsolete, such as algorithms and strategies employed by security systems, browsers, email providers, and other entities to detect and block phishing attacks. The limitations found in our literature review emphasize that phishing attempts are still a danger and that detection techniques are continually being developed.

Table 1. Evolution of phishing attacks.

Year/Period	Phishing Technique	Description	Reference
Early 2000s	Basic Email Phishing	Generic emails asking for personal details with little personalization.	[7]
Mid-2000s	Spear Phishing	Targeted attacks using personal or organizational information to increase credibility.	[8]
2010s	Whaling	Phishing aimed at high-profile individuals like executives to gain sensitive information.	[9]
2015-Present	Business Email Compromise (BEC)	Sophisticated impersonation of colleagues or vendors to fraudulently request payments.	[10]
2020s	AI-Enhanced Phishing	AI-driven phishing campaigns with highly personalized content and advanced deception techniques.	[11]

EL can effectively utilize all the information from the text-based features of ML algorithms. Among these algorithms is the one with the best performance using text-based features and the highest classification results. As a result, this improves phishing email detection performance. In light of the aforementioned observation, the objective of this paper is to tackle the following research question: Does the amalgamation of EL with text-based attributes enhance the comprehensive efficacy of detecting phishing emails? This question results from the growing number of phishing attempts that threaten the cybersecurity of individuals and organizations. Phishing emails are always changing, even with advancements in detection techniques; thus, it is critical to create more reliable and effective detection systems. To answer this question, this paper combines the use of two EL techniques, weighted and soft voting EL, with five different machine learning classifiers on textual phishing emails. Several techniques are used, such as email parsing, hashing, and feature selection. The proposed effort seeks to improve the accuracy, speed, and cost-effectiveness of predicting outcomes in phishing email detection by properly categorizing phishing emails and using all the information they include. We can convert the email into a concise but comprehensive representation by utilizing text-based features extracted from an email's message body text. EL methods can effectively determine this representation to accomplish a high level of detection performance.

The rest of this paper is structured as follows: Section 2 presents related works, encompassing similar methods employed for identifying email phishing, and scrutinizes their limitations. Section 3 offers an elaborate explanation of the proposed technique. Section 4 displays the experiments and results. Section 5 concisely outlines the key facets of the research.

2. Related Work

This section on related work presents an extensive analysis of previous research endeavors that have focused on the identification of phishing emails using machine learning methodologies. As machine learning algorithms are unable to directly comprehend the contents of emails, it becomes necessary to use feature engineering to transform an email into a concise, understandable representation. Previous research investigations have mostly focused on text-based characteristics to distinguish emails classified as spam from those that are not. Consequently, the relevant material has been classified into three separate segments. The initial segment predominantly centers on the recognition of phishing emails by employing textual characteristics. Subsequently, we investigate the utilization of ensemble classification in the domain of phishing email identification. This segment concludes by discussing the constraints of the current body of literature and clarifying how our research efforts have effectively addressed these limitations.

2.1. Identification of Phishing Emails Employing Analysis of Textual Characteristics

A notable discovery is that earlier studies (before 2015) on detecting phishing emails generally emphasize content-based characteristics, but more current studies prioritize textual characteristics. The current advancement of artificial intelligence (AI) and NLP techniques has significantly improved our ability to identify phishing emails accurately. This progress is mostly due to the use of textual information included inside the emails. In a recent publication by [12], the authors employed the TF-IDF approach to generate text-based features, followed by the utilization of a Graph Convolution Network for email categorization.

The study conducted by Radev et al. [13] involved the analysis of a total of 3685 phishing emails and 4894 benign emails. By employing 3-fold cross-validation, the study determined the accuracy rate to be 98.2%. However, the study by Alhogail [12] presents two main limitations. Firstly, number of phishing emails is significantly lower in comparison to benign ones. Secondly, the authors should have considered a more realistic assessment that involves the deployment of fresh phishing emails. The unique features of the body text of phishing emails were investigated by Gualberto et al. [14]. Using different feature combinations and machine learning methods, the researchers conducted several trials. With a success percentage of 99.95%, the XGBoost algorithm produced the best outcomes. Gualberto et al. [14] introduced a multi-stage method for spotting phishing emails, building on their work. This method combines two techniques: feature extraction and feature selection. Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) were employed in the feature extraction phase, while chi-squared statistics and Mutual Information were used in the feature selection process. Utilizing Latent Semantic Analysis (LSA) in conjunction with the XGBoost (version number 2.1.2) algorithm provided the highest level of efficacy. The same dataset was used in both studies to evaluate their approaches, yielding similar findings. However, it is crucial to recognize that both investigations need to have considered the increasing quantity of phishing emails, considering that the phishing emails under scrutiny were out of date.

A framework called THEYMIS was created in the study by Fang et al. [15] with the goal of recognizing phishing emails. This framework makes use of Recurrent Convolutional Neural Networks (RCNN). Word2Vec, along with character and word levels, were used to separate the emails. Utilizing [16], vector sequences were derived from these levels. Using the IWSPA 2018 dataset, THEMIS achieved a 99.848% detection accuracy and a 0.043% false positive rate. In a similar vein, Hiransha et al. [17] applied a deep learning (DL) technique to the identification of phishing emails. The authors used a combination of Keras Word Embedding and Convolutional Neural Networks to achieve this. This method's evaluation was carried out using the IWSPA 2018 dataset, accounting for emails with and without headers. The results of the study show that the model performs better when the email headers are ignored, with an accuracy rate of 96.8%. A common limitation between Hiransha et al. [17] and Fang et al. [15] is that they could have evaluated their models' performance on more recent emails to gauge how well they could identify the evolving nature of phishing email attacks. Moreover, Hiransha et al. [17] used an unbalanced sample ratio and improperly depended just on the accuracy measure. The SAFE-PC method uses features found in text to recognize phishing emails. Natural language processing (NLP) approaches including synonym substitution, Freebase, and Named Entity Recognition (NER) are used to generate these features. The RUS-Boost approach, which combines boosting and data sampling techniques with other machine learning algorithms, is used in the classification process. Using this method, 71% of the phishing emails that Sophos had previously incorrectly classified were properly recognized. It also achieved a 15% false positive rate (FPR). Twenty-six features were retrieved by Egozi and Verma [18] by taking word counts, punctuation, and stopwords into account. Then, different machine learning classifiers used these attributes as inputs. With a true positive rate (TPR) of 83% and a true negative rate (TNR) of 96%, Support Vector Machine (SVM) performed the best. The main drawback of Egozi and Verma [18] is that their feature sets are weak and easily manipulated because they mainly rely on terminating words, punctuation counts, word variations, and feature ratios. The utilization of Recurrent Neural Networks (RNNs) by the authors Halgavs et al. [19] enabled the identification of phishing emails through the utilization of text-based features. The RNN classifier's efficacy was evaluated against the "text analysis" model. The authors' inadequate disclosure of the phishing emails used in the study is the source of this approach's flaws. As a result, the technique is unaffected and continues to be able to identify phishing emails in real time. Additionally, it is still unclear if the authors tested the method using actual use case data that was unbalanced.

Unnithan et al. [20] employed domain-level attributes, encompassing the most frequently occurring terms and a list of special characters, as well as text-based features acquired through the utilization of the TF-IDF approach. Among the various classifiers evaluated, the Logistic Regression classifier achieved the highest F1 score of 98%. There are two limitations of this investigation. Firstly, the dataset employed necessitates updating, thus hindering the accuracy and relevance of the obtained outcomes. Secondly, the vulnerability of the domain-level characteristics utilized in the technique renders them susceptible to being easily bypassed, thereby diminishing the strategy's overall effectiveness.

In a separate investigation, Unnithan et al. [21] conducted a comparison between the TF-IDF approach and Doc2Vec, a word-embedding methodology derived from Word2Vec, to ascertain their respective efficacy in extracting text-based attributes from emails. Through experimentation with various machine learning techniques, the authors concluded that the utilization of Doc2Vec leads to improved detection performance. However, a notable drawback of this study is the possibility that the achieved accuracy of 88.4% may be higher when compared to similar studies conducted in this field.

2.2. Ensemble Learning Classification

Ensemble learning involves the integration of many Machine Learning algorithms to address a learning issue, such as classification. Ensemble learning models generally exhibit an improved capacity for generalization compared to standard ML models by integrating various algorithms [22]. Ensemble learning methods may enhance the predicted accuracy of a classification problem. During classification tasks, an ensemble learning model utilizes many algorithms for predicting the sample class. A final forecast is produced by fusing these individual algorithm predictions using techniques like voting, bagging, and weighting. In the realm of phishing website identification, the application of ensemble learning approaches has been thoroughly investigated [23]. However, these methods are still in the early stages of development when it comes to phishing email detection text. There is research in the literature that looks at how effectively ensemble learning detects phishing emails. In their investigation, the aouthors in [24] introduced an ensemble learning technique for classifying phishing emails. This approach utilizes text-based characteristics, the TF-IDF method, and syntactic features. An assessment was conducted to determine the most effective ensemble learning techniques. The study found that combining the AdaBoost and bagging methods enhances classification performance.

2.3. Limitations of the Literature

Despite recent progress in predicting the performance of phishing email detection, numerous unaddressed constraints still need to be addressed in the literature. These include the limitation of comprehensive evaluation metrics and samples, a lack of experimentation that takes into account their evolving nature, and the use of non-representative textual features. One notable limitation of the research is the need for complete consideration of various data sources. Specifically, previous studies used only a few datasets to conduct their evaluations. Although ensemble learning has achieved impressive outcomes in several fields [25], it has yet to be used in phishing email detection. Specifically, a limited number of previous studies [23,24] use ensemble learning techniques, which have many drawbacks. Some of the issues with the assessment datasets include a limited number of outdated phishing emails, inadequate representation of the emails due to insufficient implemented features, and poor evaluation measures, namely the AUC and accuracy. Therefore, there is a significant deficiency in the suitability of ensemble learning for identifying contemporary phishing email assaults. We have devised a method that applies text-based features to deliver a comprehensive email representation tailored to machine learning classification applications. The inadequacies in the literature that have been discovered are effectively addressed by this strategy. Our proposed solution uses two ensemble learning methods, weighted ensemble learning (Method 1) and soft voting ensemble learning (Method 2), which are both methods that fall under the general category of ensemble learning.

This work offers new features for evaluating our proposed method, guaranteeing a neutral and precise assessment. The primary objective is to establish a strong and efficient approach to combat the growing threat of email phishing attacks.

3. Proposed Approach

The methods for implementing the suggested approach are shown in this section. These include gathering datasets and using them for a hybrid approach that is being trained and tested to identify emails with phishing content. The overall structure of this study is shown in Figure 1, which covers feature extraction, dataset acquisition, data preparation, training, and the testing of different hybrid techniques. The research technique is described in the following subsections.

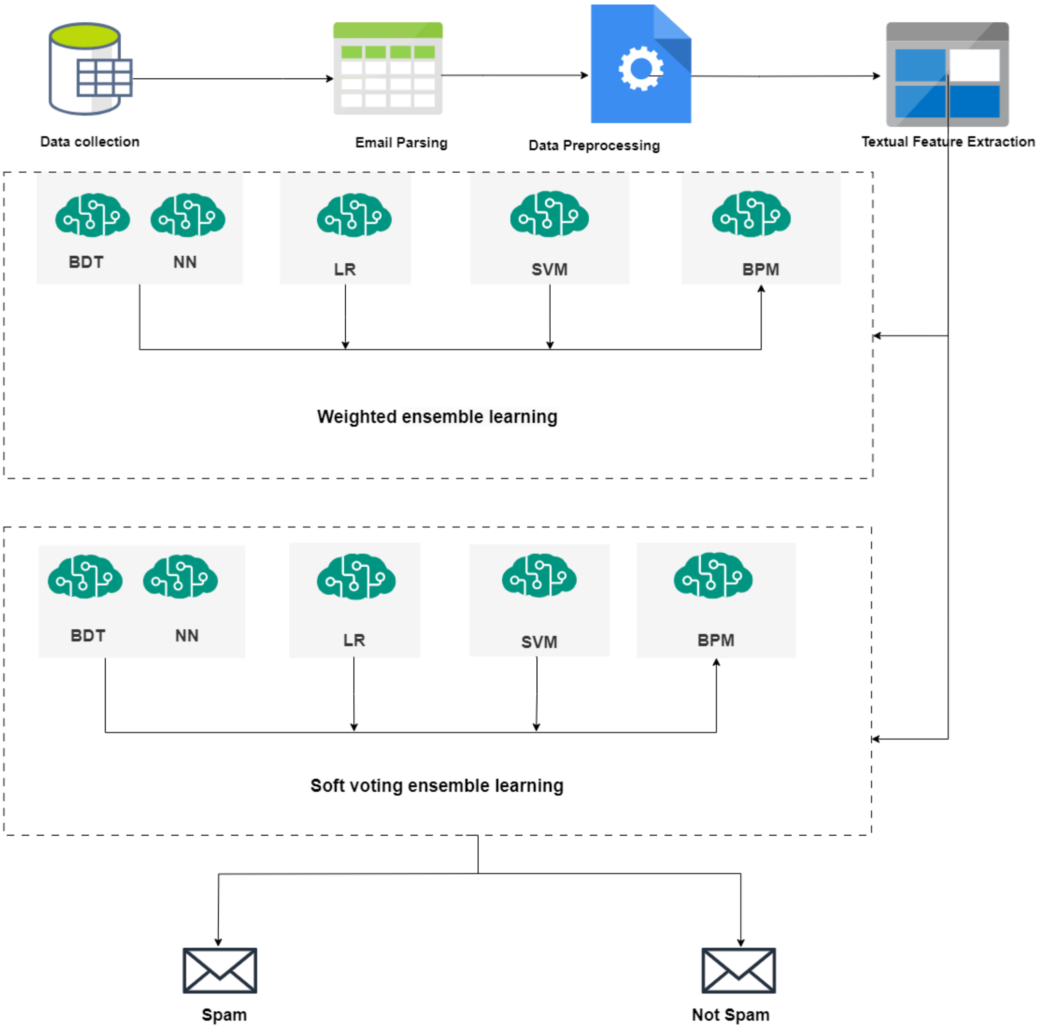

Figure 1. Methodology for phishing email detection.

3.1. Data Collection

To enhance the resilience of our model, we implemented a focused strategy to handle the textual data known as the "Spam or not spam dataset" [26]. Our augmentation efforts focused on this dataset, which consists of 5000 non-spam (ham) emails and 1000 spam emails.

Regarding this written context, our methodology focuses on using careful data augmentation techniques to effectively increase the size of this dataset. This augmentation approach purposefully uses TensorFlow to amplify the variety and complexities of textual characteristics inside this particular dataset [25]. To broaden the range of training data for natural language processing (NLP) tasks, random insertion—a text data augmentation approach—was utilized, where word-level augmentation allows us to preserve the semantic structure of phishing emails. This technique entails adding random words to the source text. The model can more effectively generalize to various text patterns if the input data are diverse. After this augmentation, a final dataset with 15,000 non-spam and 3000 spam emails was produced. Data augmentation is a technique that involves creating more examples in a training set by generating modified versions of existing data. For our experimentation, we utilized a technique called random insertion. This method involves inserting random words into the original text as a means of augmenting the data. This specific method is frequently used to artificially increase the variety of the training data for tasks associated with natural language processing (NLP). By incorporating random words, the model is exposed to variances in the input data, which can potentially improve its ability to generalize to different text patterns. Data augmentation plays a crucial role in improving the accuracy of a model by changing the data. Data augmentation is a technique that involves creating additional examples in a training set by generating modified copies of existing data.

3.2. Email Parsing

The process of extracting relevant information from emails and converting it into an organized format that computer systems can easily use and understand is known as email parsing. This procedure is especially helpful when dealing with large numbers of emails or when combining email data with databases and other applications. Email content must be broken down into its parts, such as the sender, recipient, topic, date, and message body, during the parsing process. Additionally, it may also include the extraction of certain data points, like order numbers, dates, names, or any other information pertinent to a given application [27]. Email parsing, in the context of phishing, is the process of extracting relevant data from each email, such as the body content (B), the header information (H), and the label indicating if the email is phishing or not (L).

3.2.1. Body Matrix B

In Equation (1), matrix B represents the body fields of each email.

$$B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{1}$$

3.2.2. Header Matrix H

In Equation (2), matrix H represents the header fields of each email.

$$H = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_n \end{bmatrix} \tag{2}$$

3.2.3. Label Matrix L

In Equation (3), matrix L represents the labels of each email.

$$L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_n \end{bmatrix} \tag{3}$$

3.2.4. Joint Matrix E

The last step in the email parsing process is to combine arrays B, H, and L to create a cohesive array. Both the stage of preprocessing, which gets the email body texts ready for the extraction of text-based features, and the stage of extracting content-based features, which pulls features from the email fields in the header and body, use array E. Equation (4) shows matrix E.

$$E = \begin{bmatrix} b_1 & h_1 & l_1 \\ b_2 & h_2 & l_2 \\ b_3 & h_3 & l_3 \\ \vdots & \vdots & \vdots \\ b_n & h_n & l_n \end{bmatrix} \tag{4}$$

3.3. Data Preprocessing

The process of cleaning, transforming, and arranging raw data into a format suitable for training and assessing machine learning models is known as data preprocessing, and it is a crucial step in the machine learning pipeline. The correct application of data preparation techniques can have a major impact on the effectiveness and performance of a machine learning system. One of these methods is to deal with missing data by removing or imputing missing values. For this, a variety of techniques can be used, including mean imputation, median imputation, and more complex methods like regression imputation. Another essential component of data preprocessing is data cleaning, which entails removing errors, inconsistencies, and outliers. This could include getting rid of duplicates, fixing typos, and handling outliers that could negatively affect the model itself. Additionally, data transformation—which transforms categorical variables into a format appropriate for machine learning models—is a crucial component of data preprocessing. For nominal variables, this conversion procedure may include label encoding or one-hot encoding, and for ordinal variables, ordinal encoding [28,29].

3.4. Feature Extraction

It is necessary to extract the features from the textual dataset since it contains two unique features:

- Email.
- Label.

To achieve this goal in our experimental investigation, we utilized the feature hashing (FH) technique. FH is a method that, without any linguistic analysis or preprocessing, turns unique tokens into integers. As an example, let us look at a string of short phrases like "I loved this book, I hated this book, I love books". FH builds an n-gram dictionary internally. In the aforementioned example, the list of bigrams would include phrases like "This book" (frequency: 2 times), "loved" (frequency: 1 time), "love" (frequency: 1 time), and so on. Once the aforementioned dictionary is created, FH assigns hash values to the terms and determines the utilization of each feature.

For every row of text data, the module generates a set of columns, where each column represents a hashed feature. A value of 0 in a column indicates the absence of the hashed feature in the corresponding row, while a value of 1 signifies its presence. The advantage of using feature hashing lies in its ability to represent text documents of varying lengths as numeric feature vectors of equal lengths, thereby achieving dimensionality reduction. On the contrary, if the text columns were used in their original form, they would be treated as categorical feature columns with numerous distinct values. Additionally, having the outputs in numeric form enables the utilization of various machine learning techniques, such as classification, clustering, and information retrieval. The use of integer hashes instead of string comparisons during lookup operations also leads to the faster retrieval of feature weights.

In our experiment, we utilized N-grams with a parameter value of 2, resulting in a large dataset comprising 65,539 hashed features. To effectively manage this high-dimensional dataset and prioritize the most predictive variables pertinent to phishing detection, we implemented chi-squared feature selection methodologies suitable for categorical data within classification frameworks. This particular feature selection algorithm evaluates the correlation between each feature and the label variable, thereby facilitating the identification of features that possess the most significant discriminatory capabilities. After the application of this technique, we successfully decreased the feature count to 50, thereby enhancing computational efficiency while preserving the most salient features. Furthermore, the selected features—including word frequency patterns, the presence of specific keywords, and the structure of emails—are paramount in differentiating phishing emails from legitimate communications, as they encapsulate essential characteristics of phishing attempts. By concentrating on these features, we ensure that the model not only enhances its performance but also addresses the fundamental elements of phishing behavior, thereby rendering detection more precise and pertinent to practical applications.

3.5. Model Selection

During the model selection phase of our machine learning experiment, we extensively evaluated and compared the performance of five unique models that were specifically designed to tackle our problem. The chosen models encompass a diverse array of methods, each possessing distinct strengths and qualities. Two-Class Boosted Decision Tree utilizes these functionalities.

The use of ensemble learning is discussed in the study by [30], while the Two-Class Neural Network leverages the complexity and flexibility of neural architectures, as highlighted in the study by [31]. The Two-Class Support Vector Machine utilizes a resilient classification methodology [32], while Two-Class Logistic Regression provides a straightforward yet comprehensible model. The Two-Class Bayes Point Machine (BCPM) is a classification model that functions inside the Bayesian framework. This model is specifically developed for binary classification challenges. This model, specifically tailored for binary classification tasks, is an extension of the traditional Support Vector Machine (SVM). The BCPM utilizes Bayesian concepts to estimate the parameters of the model and produce probabilistic predictions [33]. By conducting a thorough examination of performance metrics, cross-validation results, and computational factors related to each model, our objective was to determine the most appropriate candidate for our machine learning task. This allowed us to make an informed and data-driven choice during the model selection process.

3.6. Model Training

Training a machine learning model entails instructing the algorithm to distinguish between authentic emails and phishing emails using a predetermined set of criteria. A conventional approach involves employing a training dataset, which is then partitioned into distinct training and testing sets, to assess the model's performance. Regarding the dataset division, the original dataset included a total of 18,000 emails, with 15,000 classified as non-spam and 3000 as spam. For evaluation, we utilized 70% (12,600 emails) of the data as a training set, while the remaining 30% (5400 emails) were employed as a testing set. To improve the model's accuracy and its capacity to generalize across various text patterns, data augmentation approaches were utilized. The purpose of incorporating random phrases in the emails was to diversify the training data, hence enhancing the model's ability to detect phishing patterns, including potential variants. Data augmentation is an essential component in the domain of machine learning, particularly when dealing with datasets that have a restricted size. By incorporating random phrases, the model is exposed to a broader array of situations, which could potentially improve its ability to identify phishing attributes in actual email communications. This technique mitigates the risk of overfitting and improves the model's capacity to generalize effectively. After expanding the dataset, the next step was to train the phishing classifier [34].

In the previous section, a set of five distinct machine learning classifiers were utilized, which included ensemble learning-based Two-Class Boosted Decision Tree, Two-Class Neural Network, Two-Class Support Vector Machine, Two-Class Logistic Regression, and lastly, Two-Class Bayes Point Machine (BCPM).

Throughout the training phase, the model can differentiate between legitimate and phishing emails by discerning patterns and features present in the input data. This process plays a crucial role in equipping the model with the capacity to make informed decisions when faced with novel, unseen data.

3.7. Ensemble Classification

The core element of the proposed methodology is the ensemble classification phase, as illustrated in Figure 1. This phase specifically entails the binary classification of electronic mail samples to determine their categorization as unsolicited messages. The primary aim of the phase of ensemble classification is to effectively combine characteristics to improve the classification performance the proposed approach. To accomplish this, the approach relies on the use of ensemble learning as opposed to a single classifier. Ensemble learning is a technique that integrates the outcomes of multiple algorithms of machine learning, referred to as base learners, to generate a prediction accuracy that is more consistent, robust, and precise in comparison to individual base learners. The course of action of ensemble learning entails post-processing of the outcomes of the prediction of base learners to obtain the final prediction of categorization. To enhance the effectiveness of the proposed approach, two well-established techniques of ensemble learning, specifically weighted ensemble learning (Method 1) and soft voting ensemble learning (Method 2), were employed. These techniques were chosen based on their proficiency in handling various types of algorithms of machine learning and allowing the independent processing of hybrid characteristics through the use of distinct algorithms of machine learning. For instance, one algorithm can be utilized for the processing of characteristics based on content, while another algorithm can be employed for the processing of characteristics based on text. The primary distinction between Method 1 and Method 2 lies in their respective approaches to combining the outcomes of base learners and determining the outcome of classification.

The proposed approach introduces an innovative method by utilizing techniques of ensemble learning, which consists of two base learners, to effectively deal with hybrid characteristics. More specifically, the set of hybrid characteristics consists of two subsets: characteristics based on content and characteristics based on text. Each base learner independently processes these subsets concurrently, with the first base learner handling characteristics based on content and the second base learner handling characteristics based on text. This method offers the advantage of utilizing algorithms of machine learning that achieve the highest classification performance for each subset of characteristics as base learners. Consequently, the model of combination analyzes the outcomes of base learners to determine the final prediction of categorization. The selection of base learners for Method 1 and Method 2 is a complex undertaking due to the extensive range of algorithms of machine learning that are suitable for binary classification.

The proposed approach focuses exclusively on techniques of machine learning and deep learning that have previously been utilized in the identification of fraudulent emails, which have demonstrated commendable performance. A collection of machine learning algorithms was mainly established to restrict the potential choices, consisting of the following:

- Function-based Logistic Regression (LR).
- Support Vector Machine (SVM).
- Bayes Point Machine (BPM).
- Neural Network (NN).
- Boosted Decision Tree (BDT).

3.7.1. Method 1: Weighted Ensemble Learning

Weighted ensemble classification is a learning technique in which numerous basic classifiers are merged to create a more powerful and resilient classifier. Each base classifier in a weighted ensemble is allocated a weight corresponding to its significance in the ultimate decision-making process. The weights dictate the influence of each base classifier on the final prediction.

As shown in Figure 1, Method 1 assigns distinct weights to each model instead of giving each model's forecast the same weight based on each model's performance. The concept of a weighted ensemble is to provide more significance to the classifiers that demonstrate superior performance in certain parts of the data or tasks. This enables the ensemble to exploit the advantages of individual classifiers and alleviate their shortcomings, leading to a more precise and dependable aggregate prediction.

3.7.2. Method 2: Soft Voting Ensemble Learning

Soft voting ensemble learning (Method 2) (referred to as Soft Voting Classification in Figure 1) incorporates the class prediction probabilities associated with each base learner and integrates these predictions through averaging in order to ascertain the final classification. Ensemble learning typically employs two voting techniques: (a) hard voting [35] and (b) soft voting [3]. These methods differ in their approach to handling outcomes within the combination model. The Majority Voting method employs the outputs of the base learners to determine the final classification by considering the hard votes. Conversely, the soft voting method calculates the average decisions made by each base learner to determine the final classification. In comparison to Majority Voting, the soft voting method mitigates the impact of structural sensitivity caused by the base learners, reduces the variance of the ensemble, and gains additional information.

Figures 2–6 show the ROC Curves for different models used. As depicted in Figure 1, Method 2 employs two base learners, specifically the DT algorithm for content-based features and the KNN method for text-based features (Tier-1 in Figure 3). The outcome of both fundamental learners is the probability that an email belongs to a specific category. In Tier 2, these probabilities are used as inputs to an Argmax function in order to determine the most probable class and perform the final classification. To be more precise, the task of detecting phishing emails can be conceptualized as a binary classification problem.

3.8. Evaluation Metrics

Machine learning and statistics use confusion matrices to evaluate the performance of classification models. By summarizing how well a model has classified instances, the model can provide a clear picture of how well it has classified instances [36]. An actual class label and predicted class label of a classification problem are organized in a performance matrix. True positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*) represent the four main components of the performance matrix.

In classification tasks, the following performance metrics are commonly used [37]:

1. The accuracy of a classifier is measured by calculating the percentage of correctly classified instances (true positives and true negatives) divided by the total number of instances (5).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

2. Precision: Out of all occurrences anticipated to be positive (including true positives and false positives), precision is the percentage of accurately predicted positive cases (true positives) (6). This parameter measures how well the classifier avoids false positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

3. Recall (Sensitivity): Recall, also known as sensitivity or true positive rate, is the percentage of real positive cases (also known as true positives and false negatives) that the classifier properly classifies (7). It evaluates how well the classifier can identify instances of success.

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \tag{7}$$

4. F1 Score: The harmonic mean of recall and precision is the F1 score. The performance of the classifier is balanced by combining precision and recall into a single statistic. When there is an imbalance between the positive and negative cases in the dataset, it is especially helpful (8).

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

These metrics are obtained from the confusion matrix, which summarizes the predictions made by the classifier about the labels that represent the ground truth.

4. Experiments and Results

We utilized the advanced features of Microsoft Azure to create a robust system for identifying phishing emails in our experiment, and due to its advantage in speedup in the training or scalability of the proposed model. By leveraging Azure's cloud-based infrastructure, we effectively utilized scalable and reliable computing resources, enabling the efficient execution of our machine learning models. The implementation of Azure not only optimized the deployment of our models, but also provided significant information about the performance of the models and the utilization of resources. An important benefit of using Azure for detecting phishing emails is the opportunity to leverage advanced machine learning tools and services that improve the accuracy and scalability of our models. Furthermore, we utilized SQL on Azure to implement ensemble classification, which offered a robust and effective solution for database management. The incorporation of SQL into Azure not only simplified the process of storing and retrieving data, but also allowed for the smooth coordination of ensemble classification algorithms, thus enhancing the overall success of our experiment.

4.1. Evaluation Approach

Evaluating ensemble classifiers is a crucial aspect of understanding their effectiveness and suitability in various areas. Prior studies have explored the assessment of ensemble techniques, shedding light on their effectiveness across different datasets and problem domains. Ensemble classifiers, which amalgamate the predictions of multiple foundational models to augment overall accuracy and resilience, have garnered considerable attention in the realm of machine learning literature. Several studies show limitations in genuine detection when employing ensemble phishing categorization, wherein research papers fail to provide insights into the methodology of detecting ensemble phishing emails [38,39]. To circumvent these constraints, we adhered to the following guidelines that were taken into consideration to augment the resilience of the proposed approach and forestall the acquisition of deceptive outcomes.

- Comprehensive Ensemble Composition Description:
  We ensured detailed and transparent presentation of the composition of our ensemble classifiers. We describe the underlying classifiers used and how they are configured. This transparency is critical for attaining replication and makes it easier to understand how the ensemble uses the unique capabilities of individual classifiers in the context of phishing email detection.
- Realistic Phishing Email Sample Deployment:
  We deployed realistic phishing email samples to ascertain the adaptability and effectiveness of our ensemble detection system in identifying contemporary phishing threats.
- Addressing Imbalanced Class Ratios Appropriately:
  We conducted evaluations of our ensemble classifiers using imbalanced class ratios to simulate real-world scenarios where benign emails substantially outnumber phishing ones. This guideline guarantees the robustness and efficacy of our ensemble approach in dealing with the practical challenges associated with class imbalance in phishing detection.
- Metric Selection Aligned with Class Imbalance:
  Metrics such as precision, recall, and F1 score prove more suitable for imbalanced datasets as compared to metrics influenced by class distribution, such as accuracy [40].
- Feature Importance Analysis:
  Many studies have often neglected the examination of feature importance in ensemble classifiers for phishing email detection. By comprehending the importance of different features, researchers can gain insights into the underlying patterns that contribute to effective phishing detection and enhance interpretability.

4.2. Dataset

Our research aims to gather comprehensive and practical data to improve the accuracy of our ensemble model in identifying phishing attempts. In order to achieve this, we obtained data from a range of publicly available databases. The collection comprised authentic and fraudulent emails obtained from Apache Spam Assassin's datasets. Our research aimed to enhance the durability of our phishing detection model by specifically augmenting the data. The dataset used comprises 5000 legal emails and 1000 fraudulent emails. In order to increase the dataset's diversity and improve the model's ability to generalize, we utilized data augmentation techniques with TensorFlow. To be more precise, we utilized a method called random insertion, which entails incorporating arbitrary words into the original text. This strategy exposes the model to several forms of input data, hence improving its capacity to identify a wide range of text patterns.

In order to comply with the specified requirements, an uneven allocation of classes was put into effect. More precisely, a ratio of 1:5 was used to categorize phishing and benign emails, indicating that phishing emails account for around 5% of the benign samples. To conduct a thorough study, the dataset was partitioned into two subsets: one for training and validation, and the other for testing the trained models. The training and validation subset consists of 70% of the samples, while the remaining 30% is reserved for testing. The training set has 10,500 samples, whereas the testing set contains 4500 samples. The sets are divided into 7350 benign samples and 3150 phishing samples for training/validation, and 3150 benign samples and 1350 phishing samples for testing.

Table 2 provides a comprehensive overview of the number of email samples present in the training and testing sets, as well as the cumulative total.

Table 2. The details of the evaluation dataset.

	Benign	Phishing	Total
Training set	7350	3150	10,500
Testing set	3150	1350	4500
Total	10,500	4500	15,000

4.3. Selection of Base Learners

The process of selecting the machine learning algorithms to serve as base learners presents a substantial challenge during the phase of ensemble learning. During this task, the machine learning algorithms are assessed based on distinct feature sets in order to ascertain which algorithm offers optimal efficiency. We conducted evaluations on various renowned machine learning algorithms (LR, SVM, BPM, NN, BDT), utilizing the evaluation metrics that were previously discussed in Section 3.8. The results are presented in Table 3.

Table 3. Evaluation of text-based features.

Classifier	F1-Score	Accuracy	Precision	Recall	FP	FN	TN	TP
LR	0.757	0.934	0.979	0.617	2	57	748	92
SVM	0.844	0.951	0.895	0.799	14	30	736	119
BPM	0.870	0.961	0.975	0.785	3	32	747	117
NN	0.880	0.963	0.960	0.812	5	28	745	121
BDT	0.905	0.969	0.912	0.899	13	15	737	134

Regarding the results of the text-based features, BDT achieved an impressive F1-score of 0.905, showcasing its ability to balance precision and recall. This is crucial in scenarios with an imbalance between the classes, as the F1-score considers the false positive and false negative rate results. NN demonstrated the second-best F1-score of 0.880, indicating strong performance in capturing both precision and recall. However, it is slightly lower than BDT, highlighting the latter's superiority in this particular evaluation metric. Logistic Regression

(LR) achieved an F1-score of 0.757, indicating a moderate balance between precision and recall. Its accuracy is relatively high at 0.934, with a precision of 0.979, suggesting a low rate of false positives. However, its recall is 0.617, indicating that LR may not capture all positive instances.

Support Vector Machine (SVM) demonstrated improved performance, with an F1-score of 0.844, reflecting a better balance between precision and recall compared to LR. Its accuracy is 0.951, and the precision is 0.895, indicating a relatively low false positive rate. Its recall, at 0.799, suggests that SVM effectively identifies a substantial portion of positive instances.

Bagging with a Passive-Aggressive Model (BPM) achieved an F1-score of 0.870, indicating balanced performance. Its high accuracy (0.961) and precision (0.975) demonstrate its capability to minimize false positives. However, its recall (0.785) suggests a trade-off with false negatives.

Neural Network (NN) performed well, with an F1-score of 0.880, showcasing a good balance between precision and recall. Its accuracy (0.963) and precision (0.960) are high, emphasizing effective positive instance identification. Its recall (0.812) suggests a reasonably low false negative rate.

Boosted Decision Tree (BDT) demonstrated the highest F1-score at 0.905, indicating strong overall performance. Its accuracy (0.969) and precision (0.912) are also noteworthy, with a high recall of 0.899, suggesting the effective identification of positive instances. Boosted Decision Tree (BDT) stands out as the top-performing classifier among those evaluated for text-based features, with a high F1-score, accuracy, precision, and recall. To clarify our results, Figures 2–6 provide ROC (Receiver Operating Characteristic) curves, which are a valuable tool for assessing the trade-off between sensitivity and specificity in a binary classification model, particularly in scenarios where the class distribution is imbalanced. The ROC curve plots the true positive rate (sensitivity) against the false positive rate at various threshold settings. Each point on the ROC curve represents a different threshold for the binary classifier [41].

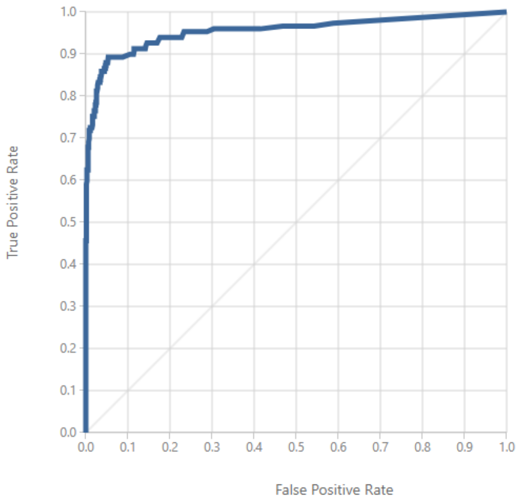

Figure 2. ROC Curve for Logistic Regression.

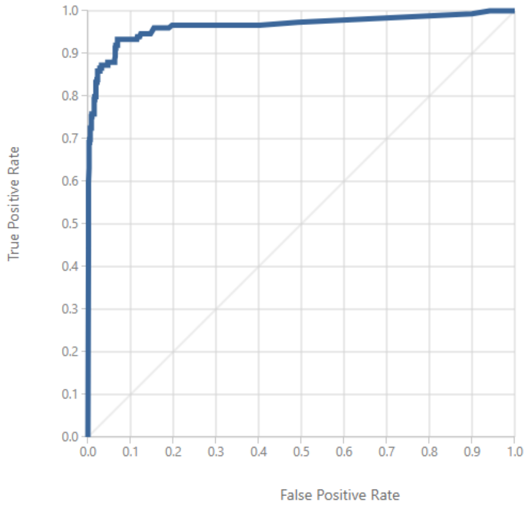

Figure 3. ROC Curve for Support Vector Machine.

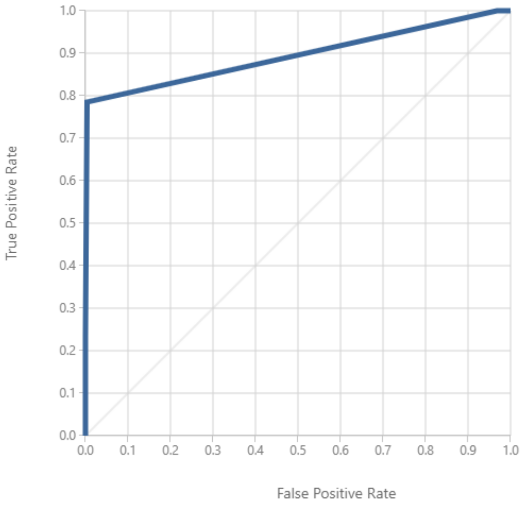

Figure 4. ROC Curve for Bayesian Probabilistic Model.

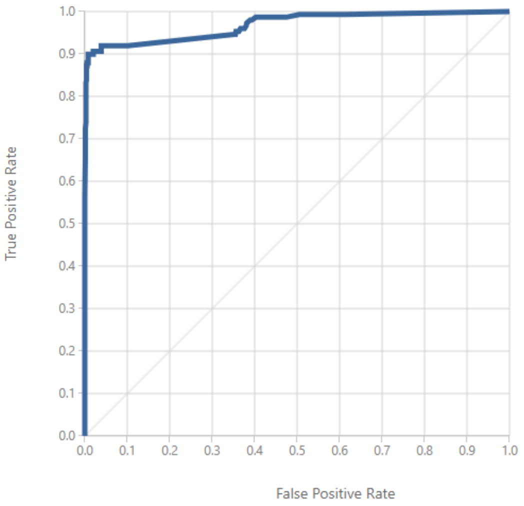

Figure 5. ROC Curve for Decision Tree.

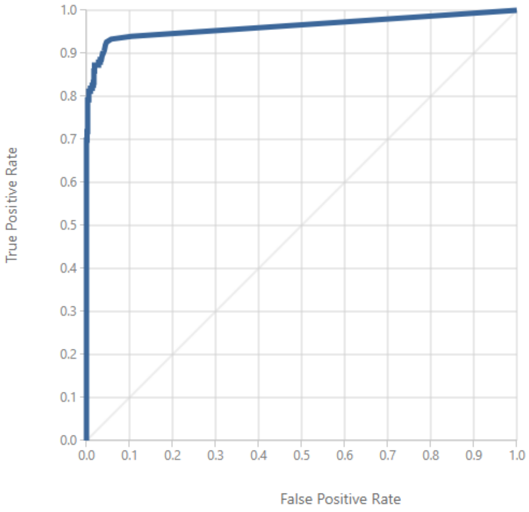

Figure 6. ROC Curve for Neural Network.

4.3.1. Method 1 (Weighted Ensemble)

In typical machine learning scenarios, equal treatment is given to each class during the training process. The objective of the model is to minimize a loss function that takes into account the performance across all classes. When there is a class imbalance, the assignment of different weights to each class helps to address this issue. The intention is to provide greater significance to the minority class by increasing its weight and, consequently, the contribution of its samples to the overall loss function. The class weights are integrated into the loss function utilized during training. The loss for each example is multiplied by its corresponding class weight. Consequently, errors in the minority class have a greater impact on the overall loss, guiding the model to prioritize accurate prediction of the minority class. Figure 7 presents how we used class weights to improve class imbalance.

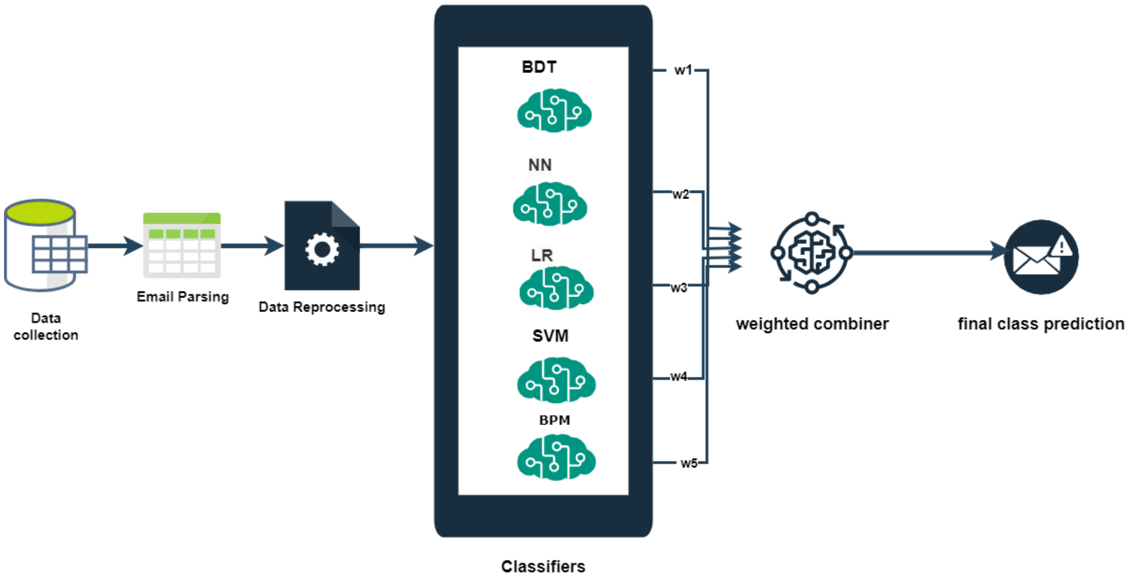

Figure 7. Weighted ensemble classifier

- Setting Class Weights: The determination of class weights is necessary to address the imbalance present in the dataset. It is customary for the weight assigned to each class to be inversely proportional to its frequency. Equation (9) is the formula for class weight:

$$w_i = \frac{N}{\text{num\_samples\_class\_i}} \quad (9)$$

where N represents the total number of samples in the dataset, and num_samples_class_i is the number of samples belonging to class i.

- Incorporating Class Weights: One approach to incorporating class weights into the training process is by modifying the loss function. Common loss functions, including cross-entropy, can be adjusted to accommodate the influence of class weights.
  A practical example of this can be seen in binary classification, where the weighted cross-entropy loss for an individual sample can be computed as follows (10):

$$L(y, \hat{y}) = -w_{\text{pos}} \cdot y \cdot \log(\hat{y}) - w_{\text{neg}} \cdot (1 - y) \cdot \log(1 - \hat{y}) \quad (10)$$

  where y is the true label (0 or 1), $\log(\hat{y})$ is the predicted probability, and w_{pos} and w_{neg} are the weights for the positive and negative classes, respectively.
- Implementation in Machine Learning Frameworks. During the process of model evaluation, we employed a Python script within the AZURE computing environment. Upon completion of the training phase, it becomes of utmost importance to evaluate the performance of the model by utilizing widely accepted metrics such as precision, recall, F1 score, or the area under the receiver operating characteristic curve (AUC-ROC). The results for the first ensemble classification method are presented in Table 4.

Table 4. Weighted ensemble classification result.

Method	F1-Score	Accuracy	Precision	Recall
Method 1 (Weighted Ensemble)	0.90	0.91	0.91	0.90

As we can see in Figure 8, method 1 demonstrates an F1-score of 0.9, a value that is competitive and close to that of the best-performing individual classifier (BDT, which exhibits an F1-score of 0.905). This observation suggests that the ensemble method efficiently captures both precision and recall, thereby achieving a favorable equilibrium between false positives and false negatives. The precision of Method 1 (0.91) is marginally lower than some individual classifiers, such as BPM (0.975) and NN (0.96). Nevertheless, the disparity is not significant, and the ensemble approach still maintains a high level of precision. Method 1 harnesses the power of an ensemble of classifiers, thereby potentially minimizing overfitting and effectively capturing diverse patterns within the data. Consequently, it tends to possess greater robustness and generalizability when confronted with unseen examples, contributing to its efficacy in detecting instances of phishing. The significance of higher precision or recall may vary depending on the specific requirements. Method 1 appears to strike a commendable balance between the two performance measures.

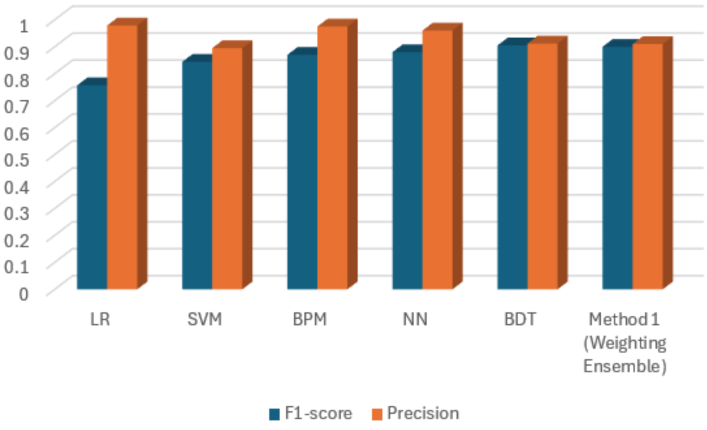

Figure 8. Phishing detection classifier performance comparison—method 1.

4.3.2. Method 2 (Soft Voting)

Soft voting is a technique used in machine learning ensembles to combine the predictions of multiple models. It involves taking a majority vote or averaging the predictions of individual models to make a final prediction. Soft voting has been applied in various domains, including offensive language detection on social media platforms [42], heart disease diagnosis [43], and predicting cyber-attacks [44]. In these studies, soft voting ensembles consisting of different machine learning algorithms were used to improve the accuracy, precision, and recall of the models. The results showed that the soft voting ensembles outperformed individual models and other ensemble methods, achieving high accuracy and area under the curve scores. Soft voting has proven to be an effective technique for improving the performance of machine learning models in various applications. Table 5 presents the soft voting results after employing our Python script in AZURE.

Table 5. Soft voting ensemble classification result.

Method	F1-Score	Accuracy	Precision	Recall
Method 2 (Soft Voting Ensemble)	0.8512	0.9556	0.9442	0.7824

Figure 9 shows that Method 2 (Soft Voting) maintains a good balance between precision and recall, which is crucial in phishing detection. Considering specific requirements, where either higher precision or recall may be more critical, Method 2 proves to be suitable for addressing these needs.

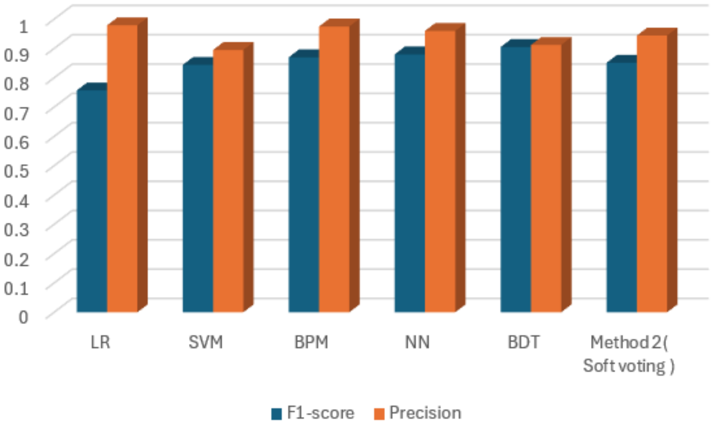

Figure 9. Phishing detection classifier performance comparison—method 2.

4.4. Discussion

This evaluation of text-based features yields valuable insights regarding the performance of different classifiers in the detection of phishing emails. Each classifier displays distinct strengths and weaknesses, underscoring the significance of selecting an appropriate algorithm based on specific requirements. BDT achieves an impressive F1-score of 0.905, thereby showcasing a robust ability to balance precision and recall. BDT stands out among the individual classifiers, as it exhibits the highest F1 score, accuracy, precision, and recall. The interpretability of Decision Tree models may raise concerns; however, in this particular context, the emphasis on informative features contributes to its success. NN performs well, attaining an F1-score of 0.880, thereby demonstrating a good equilibrium between precision and recall. NN exhibits high accuracy and precision, rendering it effective in identifying positive instances. Although NN's performance is commendable, it slightly trails behind BDT, suggesting the latter's superiority in this evaluation.

LR attains a moderate F1-score of 0.757, denoting harmonious performance in terms of precision and recall. Logistic Regression (LR) provides high levels of accuracy and precision. However, the recall value is considerably lower, suggesting a limitation in capturing all positive cases.

Support Vector Machine (SVM) demonstrates improved performance, with an F1-score of 0.844. This indicates a better balance between precision and recall compared to Logistic Regression (LR). Support Vector Machine (SVM) exhibits a comparatively low percentage of false positives, indicating its effectiveness in accurately identifying positive cases.

BPM achieves an F1-score of 0.870, demonstrating high accuracy and precision, albeit at the cost of memory. BPM efficiently reduces the occurrence of incorrect positive results, demonstrating its proficiency in handling situations when there is an unequal distribution of classes.

The trade-off associated with recollection is that BPM may fail to identify certain good examples. The weighted ensemble model demonstrates competitive performance, with an F1 score of 0.90. This strategy successfully achieves a desirable balance between precision and recall. Moreover, it possesses a short training duration, making it highly computationally efficient. Although select individual classifiers may have significantly higher precision, the overall balance is maintained.

The soft voting method achieves an F1-score of 0.8512 while also maintaining a favorable equilibrium between precision and recall. It is suited for situations where increased precision or recall is more important. Soft voting is particularly effective in the field of ensemble learning for detecting phishing.

It is important to emphasize that the choice of the most suitable classifier or ensemble approach depends on unique requirements. The BDT classifier stands out as the highest-performing individual classifier, while the ensemble techniques (Method 1 and Method 2) demonstrate competitive performance, providing a favorable trade-off between precision and recall. The inclusion of ROC curves offers further validation of this evaluation, highlighting the trade-off between sensitivity and specificity demonstrated by each classifier and ensemble approach. These comprehensive assessments provide significant insights into the benefits and considerations of each technique in recognizing phishing emails.

4.5. Comparison of Our Approach's Results with Existing Works

This section provides a detailed analysis of how our results compare to those of prominent earlier research. It is crucial to acknowledge that comparing our technique directly with existing works is difficult because each study uses different evaluation approaches, leading to a diverse range of methods. These variations include different sample sets, class ratios, and metrics. Tables 6 and 7 present the results of existing works, showcasing their performance in terms of common evaluation metrics such as accuracy and F1-score. This table also provides information about the methods used and the insights gained from these studies.

In the assessment of the efficacy of our two proposed techniques for detecting phishing emails, it is apparent that both the weighted ensemble and soft voting methodologies outperform individual classifiers and demonstrate competitive performance. By utilizing a dataset consisting of 15,000 non-spam and 3000 spam emails, our thorough analysis highlights the strengths and weaknesses of different classifiers. The individual classifiers, namely BDT, NN, LR, SVM, and BPM, exhibit diverse performance in terms of precision, recall, and F1 score. BDT stands out, with an impressive F1-score of 0.905, showcasing a resilient balance between precision and recall. While NN performs well, with an F1-score of 0.880, BDT surpasses it, indicating its superiority within this context. LR, SVM, and BPM display moderate-to-high performance in various aspects, highlighting the trade-offs between precision and recall. The ensemble methods, particularly weighted ensemble, and soft voting, demonstrate noteworthy F1-scores of 0.90 and 0.8512, respectively. The weighted ensemble approach demonstrates a notable balance between precision and recall, as well as efficient training times, indicating computing efficiency. However, soft voting achieves a balanced trade-off between precision and recall, making it appropriate for situations when either parameter is more important. Although ensemble approaches have somewhat lower precision than individual classifiers, they efficiently maintain an overall balance. Upon comparing our results with the findings reported in Tables 6 and 7, our methods demonstrate a competitive level of performance. The BDT classifier is identified as the highest-performing individual classifier, while the ensemble approaches (Method 1 and Method 2) demonstrate effectiveness in attaining a balance between precision and recall. Recognizing the variation in evaluation methods used in previous studies is essential, taking into account elements such as the sets of samples, ratios of different classes, and metrics used. Our two proposed ensemble approaches have shown encouraging results in identifying phishing emails, indicating their usefulness in achieving a balance between precision and recall. The thorough comparison with previous studies further confirms the competitive performance of our methods, highlighting the significance of taking specific criteria into account when choosing the most appropriate classifier or ensemble method for detecting phishing. The ROC curves provide further insights into the sensitivity and specificity demonstrated by each technique, offering a comprehensive perspective on their strengths and limitations in detecting phishing emails.

Table 6. Comparison of phishing detection approaches.

Reference	Insights	Results	Methods Used
[45]	The provided paper does not mention ensemble classification in phishing detection. The paper focuses on brute-force detection using ensemble classification.	The stacking algorithm achieved the highest accuracy, precision, recall, and F1. Test results: 94.87% accuracy, 99.94% precision, 98.82% recall, 99.37% F1.	Four types of ensemble classifiers were used. The stacking algorithm achieved the best test results.
[46]	The manuscript introduces a composite ensemble learning model, consisting of Decision Trees (DTree) and Naive Bayes (NBayes), to classify phishing instances. This model demonstrated superior performance compared to the hybrid model, as evidenced by the evaluation of various performance measures.	The ensemble model exhibited superior performance when compared to the hybrid model in terms of the utilized measures. In addition, the DTree and STACKING techniques demonstrated exceptional performance in comparison to the other models.	A composite learning model consisting of Decision Tree (DTree) and Naive Bayes (NBayes) was utilized. The STACKING method was employed, with DTree serving as the foundational learner.
[47]	This paper discusses the use of ensemble learning models, specifically AdaBoost, CatBoost, and Gradient Boosting Classifier, for detecting phishing websites. The best result was achieved by the AdaBoost Classifier algorithm, with an average ROC AUC score of 99%.	The AdaBoost Classifier algorithm demonstrated the best result for predicting phishing websites. The average ROC AUC score for the AdaBoost Classifier algorithm was 99%.	Ensemble learning algorithms (AdaBoost, CatBoost, Gradient Boosting Classifier). Exploratory Data Analysis (EDA).
[48]	The paper discusses the use of ensemble models, such as CatBoost Classifier, for detecting phishing attacks on web networks.	The CatBoost Classifier yielded the most optimal outcomes in terms of both accuracy and F1 measure. SHAP values, on the other hand, serve the purpose of discerning significant attributes within the model.	A variety of machine learning models were employed to identify instances of phishing. The employed models encompassed K-means, Random Forest, Decision Tree, the CatBoost classifier, the LightGBM classifier, AdaBoost, and the voting classifier.
[24]	This paper proposes an ensemble model for phishing detection that combines Decision Tree, Support Vector Machine, and Logistic Regression classifiers using the voting scheme ensemble technique.	The proposed ensemble model achieved 99.02% accuracy with 10-fold cross-validation. Machine learning-based classification techniques provided an accurate method of classification.	Decision Tree (DT), Support Vector Machine (SVM), and Logistic Regression (LR). Ensemble classifiers like Extreme Gradient Boosting (XGBoost) and Adaptive Boosting (AdaBoost).
[49]	This paper proposes a framework for phishing detection using ensemble techniques, specifically the random forest algorithm. The results show that the proposed framework outperforms other ensemble classifiers and conventional classification algorithms.	The proposed framework achieved an accuracy of 98.64%. The precision, recall, and F-measure were 0.986.	Random Forest ensemble techniques combined with k-means clustering. Wrapper-based and filter-based methods for feature selection.

Table 7. Results comparison with state-of-the-art methods.

Reference	F1-Score (%)	Accuracy (%)	Feature Category	Number of Benign Samples	Number of Phishing Samples
[12]	98.5	98.2	Text	4894	3685
[14]	99.9	99.9	Text	4150	2279
[50]	100	100	Text	4150	2279
[15]	99.33	99.84	Text	7781	999
[17]	–	96.8	Text	5088	612
[18]	99	–	Text	7689	1210
[19]	98.63	98.91	Text	6951	4572
[21]	98	97	Text	7781	997
[20]	–	88.4	Text	8913	1087

5. Conclusions

Phishing attacks, which are becoming more common in today's world, pose an increasing threat, especially with the expanding use of remote work. With the constant development of advanced techniques by adversaries to exploit weaknesses, the importance of strong cybersecurity safeguards is now more crucial than ever. This study focused on tackling one of the main methods used for such assaults—phishing emails—and presented a technique called ensemble learning to enhance their detection.

The utilization of ensemble learning, which includes both soft voting and weighted approaches, represents notable progress in the field of phishing detection. Our methodology effectively addresses the constraints found in previous studies by using two base learners to analyze characteristics. This demonstrates our in-depth understanding of the changing threat scenario. This method is significant because it can effectively identify phishing emails and is also resistant to potential disinformation.

By conducting extensive trials, we have determined the most effective machine learning algorithms to use as base learners, which guarantees a strong foundation for our ensemble approach. Our thorough assessment, following current criteria, highlights the significance of reliable and accurate results in the field of cybersecurity. Our results demonstrate the superiority of this ensemble learning strategy, which combines soft voting and weighted approaches. This approach outperforms existing techniques and achieves an impressive F1-score of 0.90 in the weighted ensemble method and 0.85 in the soft voting method.

6. Future Directions

While ensemble learning has shown significant efficacy in phishing detection by utilizing multiple classifiers, our study has some limitations that could be addressed in future work to improve its robustness and applicability. One of the main limitations is the ability of the model to generalize across different datasets. The validation of our model is currently restricted to a single dataset, which can limit its ability to adapt to various phishing techniques that are seen in practical applications. To improve the ensemble model's resistance to evolving specific phishing techniques, future research could address this issue by testing it on a wider range of datasets, sourced from other platforms and geographical regions. Additionally, utilizing synthetic data augmentation techniques could enhance the model's ability to identify novel and unusual phishing patterns. Future research could focus on integrating continuous learning frameworks too, which would enable the model to adapt to new phishing techniques without needing to be completely retrained. This strategy might be essential to preserving the model's performance in real-time applications. Finally, any potential errors in the ensemble model must be evaluated and minimized. A model may not generalize effectively to all phishing scenarios if it is trained exclusively on particular kinds of phishing data. It may be possible to create a model that is more broadly applicable to various phishing attack types.

Author Contributions: Conceptualization, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Methodology, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Software, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Formal Analysis, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Writing Original Draft, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Writing Review and Editing, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A.; Supervision, Z.S., H.A.O., E.A.E., E.A., S.A. and N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to express their sincere appreciation to Zahid Iqbal from the Department of Computer Sciences, Faculty of Computing and IT, University of Gujrat, PK, for his invaluable assistance to reacting to the reviewers' comments. His insights really improved the quality of this of research work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kalderemidis, I.; Farao, A.; Bountakas, P.; Panda, S.; Xenakis, C. GTM: Game Theoretic Methodology for optimal cybersecurity defending strategies and investments. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–9.
2. Anon. Enisa Threat Landscape 2020—Phishing. Available online: https://www.enisa.europa.eu/publications/phishing (accessed on 19 November 2023).
3. Dietterich, T.G. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
4. Das, A.; Baki, S.; El Aassal, A.; Verma, R.; Dunbar, A. SoK: A comprehensive reexamination of phishing research from the security perspective. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 671–708. [CrossRef]
5. Bountakas, P.; Xenakis, C. HELPHED: Hybrid Ensemble Learning PHishing Email Detection. *J. Netw. Comput. Appl.* **2023**, *210*, 103545. [CrossRef]
6. Opara, C.; Chen, Y.; Wei, B. Look before You leap: Detecting phishing web pages by exploiting raw URL And HTML characteristics. *Expert Syst. Appl.* **2024**, *236*, 121183. [CrossRef]
7. Stojnic, T.; Vatsalan, D.; Arachchilage, N.A. Phishing email strategies: Understanding cybercriminals' strategies of crafting phishing emails. *Secur. Priv.* **2021**, *4*, e165. [CrossRef]
8. Kwak, Y.; Lee, S.; Damiano, A.; Vishwanath, A. Why do users not report spear phishing emails? *Telemat. Inform.* **2020**, *48*, 101343. [CrossRef]
9. Gusev, A. Domestic private banking solutions can be quite successful as an effective protection against whaling-style cyber attacks which are used as a basis for more complex targeted phishing. *Procedia Comput. Sci.* **2022**, *213*, 391–399. [CrossRef]
10. Papathanasiou, A.; Liontos, G.; Liagkou, V.; Glavas, E. Business Email Compromise (BEC) Attacks: Threats, Vulnerabilities and Countermeasures—A Perspective on the Greek Landscape. *J. Cybersecur. Priv.* **2023**, *3*, 610–637. [CrossRef]
11. Chinnasamy, P.; Krishnamoorthy, P.; Alankruthi, K.; Mohanraj, T.; Kumar, B.S.; Chandran, L. AI Enhanced Phishing Detection System. In Proceedings of the 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Krishnankoil, Virudhunagar District, Tamil Nadu, India, 14–16 March 2024; pp. 1–5.
12. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
13. Radev, D. CLAIR Collection of Fraud Email, ACL Data and Code Repository. Available online: http://aclweb.org/aclwiki (accessed on 19 November 2023).
14. Gualberto, E.S.; De Sousa, R.T.; Thiago, P.D.B.; Da Costa, J.P.C.; Duque, C.G. From feature engineering and topics models to enhanced prediction rates in phishing detection. *IEEE Access* **2020**, *8*, 76368–76385. [CrossRef]
15. Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; Yang, Y. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access* **2019**, *7*, 56329–56340. [CrossRef]
16. Worth, P.J. Word embeddings and semantic spaces in natural language processing. *Int. J. Intell. Sci.* **2023**, *13*, 1–21. [CrossRef]
17. Hiransha, M.; Unnithan, N.A.; Vinayakumar, R.; Soman, K.; Verma, A. Deep learning based phishing e-mail detection. In Proceedings of the 1st AntiPhishing Shared Pilot 4th ACM International Workshop Security Privacy Analytics (IWSPA), Tempe, AZ, USA, 19–21 March 2018; pp. 1–5.
18. Egozi, G.; Verma, R. Phishing email detection using robust nlp techniques. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 7–12.
19. Halgaš, L.; Agrafiotis, I.; Nurse, J.R. Catching the phish: Detecting phishing attacks using recurrent neural networks (rnns). In *Proceedings of the Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, South Korea, 21–24 August 2019, Revised Selected Papers 20*; Springer: Cham, Switzerland, 2020; pp. 219–233.
20. Unnithan, N.A.; Harikrishnan, N.; Vinayakumar, R.; Soman, K.; Sundarakrishna, S. Detecting phishing E-mail using machine learning techniques. In Proceedings of the 1st Anti-Phishing Shared Task Pilot 4th ACM Iwspa Co-Located 8th ACM Conference Data Application Security Privacy (Codaspy), 19–21 March 2018; pp. 51–54.

21. Unnithan, N.A.; Harikrishnan, N.; Akarsh, S.; Vinayakumar, R.; Soman, K. *Machine Learning Based Phishing E-Mail Detection*; Security-CEN@ Amrita: Coimbatore, India, 2018; pp. 65–69. Available online: https://ceur-ws.org/Vol-2124/paper_12.pdf (accessed on 19 November 2023).
22. Meena, K.; Upadhyaya, S.R. A Privacy-Preserving Machine Learning Ensemble for Spam Detection. In Proceedings of the 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 3–5 August 2023; pp. 255–259.
23. Bhardwaj, U.; Sharma, P. Email spam detection using bagging and boosting of machine learning classifiers. *Int. J. Adv. Intell. Paradig.* **2023**, *24*, 229–253. [CrossRef]
24. Pathak, P.; Shrivas, A.K. Classification of Phishing Website Using Machine Learning Based Proposed Ensemble Model. In Proceedings of the 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), Chhattisgarh, India, 8–10 February 2023; pp. 1–6.
25. Zheng, C.; Wu, G.; Li, C. Toward Understanding Generative Data Augmentation. *arXiv* **2023**, arXiv:2305.17476.
26. Ozler, H. Spam or Not Spam Dataset. 2020. Available online: https://www.kaggle.com/datasets/ozlerhakan/spam-or-not-spam-dataset (accessed on 20 October 2024).
27. Kumar, A.; Chatterjee, J.M.; Díaz, V.G. A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 486. [CrossRef]
28. Cho, E.; Chang, T.W.; Hwang, G. Data preprocessing combination to improve the performance of quality classification in the manufacturing process. *Electronics* **2022**, *11*, 477. [CrossRef]
29. Maharana, K.; Mondal, S.; Nemade, B. A review: Data pre-processing and data augmentation techniques. *Glob. Transitions Proc.* **2022**, *3*, 91–99. [CrossRef]
30. Durga Bhavani, K.; Ferni Ukrit, M. Design of inception with deep convolutional neural network based fall detection and classification model. *Multimed. Tools Appl.* **2024**, *83*, 23799–23817. [CrossRef]
31. Chou, C.Y.; Hsu, D.Y.; Chou, C.H. Predicting the onset of diabetes with machine learning methods. *J. Pers. Med.* **2023**, *13*, 406. [CrossRef]
32. Roy, A.; Chakraborty, S. Support vector machine in structural reliability analysis: A review. *Reliab. Eng. Syst. Saf.* **2023**, *233*, 109126. [CrossRef]
33. Kesav, N.; MG, J. A deep learning approach with Bayesian optimized Kernel support vector machine for COVID-19 diagnosis. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2023**, *11*, 623–637. [CrossRef]
34. Nguyen, C.; Jensen, M.; Day, E. Learning not to take the bait: A longitudinal examination of digital training methods and overlearning on phishing susceptibility. *Eur. J. Inf. Syst.* **2023**, *32*, 238–262. [CrossRef]
35. Alotaibi, B.; Alotaibi, M. Consensus and majority vote feature selection methods and a detection technique for web phishing. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 717–727. [CrossRef]
36. Valero-Carreras, D.; Alcaraz, J.; Landete, M. Comparing two SVM models through different metrics based on the confusion matrix. *Comput. Oper. Res.* **2023**, *152*, 106131. [CrossRef]
37. Padilla, R.; Netto, S.L.; Da Silva, E.A. A survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
38. Dasari, S.S.P.V.; Kolukula, N.R. Improved Phishing Detection using Ensemble Models in Machine Learning. *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)* **2023**, *11*, 1401–1410. [CrossRef]
39. Subba, B. A heterogeneous stacking ensemble-based security framework for detecting phishing attacks. In Proceedings of the 2023 National Conference on Communications (NCC), Guwahati, India, 23–26 February 2023; pp. 1–6.
40. Thölke, P.; Mantilla-Ramos, Y.J.; Abdelhedi, H.; Maschke, C.; Dehgan, A.; Harel, Y.; Kemtur, A.; Berrada, L.M.; Sahraoui, M.; Young, T.; et al. Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage* **2023**, *277*, 120253. [CrossRef]
41. Flach, P.A. ROC analysis. In *Encyclopedia of Machine Learning and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–8.
42. Fieri, B.; Suhartono, D. Offensive Language Detection Using Soft Voting Ensemble Model. *MENDEL* **2023**, *29*, 1–6. [CrossRef]
43. Nazri, R.A.; Das, S.; Promi, R.T.H. Heart Disease Prediction using Synthetic Minority Oversampling Technique and Soft Voting. In Proceedings of the 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 8–9 July 2021; pp. 1–6.
44. Khan, M.A.; Iqbal, N.; Jamil, H.; Kim, D.H.; et al. An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection. *J. Netw. Comput. Appl.* **2023**, *212*, 103560. [CrossRef]
45. Kharismadhany, E.; Ruswiansari, M.; Harsono, T. Brute-force Detection Using Ensemble Classification. *INTEK J. Penelit.* **2023**, *9*, 98–104. [CrossRef]
46. Folorunso, S.O.; Ayo, F.E.; Abdullah, K.K.A.; Ogunyinka, P.I. Hybrid vs ensemble classification models for phishing websites. *Iraqi J. Sci.* **2020**, *61*, 3387–3396. [CrossRef]
47. Barabash, O.; Ziuziun, V.; Kubiavka, L. SOLVING THE PROBLEM OF DETECTING PHISHING WEBSITES USING ENSEMBLE LEARNING MODELS. *Sci. J. Astana IT Univ.* **2022**, *12*, 24–33.

48. Puri, N.; Saggar, P.; Kaur, A.; Garg, P. Application of ensemble Machine Learning models for phishing detection on web networks. In Proceedings of the 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), Sonepat, India, 8–9 July 2022; pp. 296–303.
49. Alsharaiah, M.; Abu-Shareha, A.; Abualhaj, M.; Baniata, L.; Adwan, O.; Al-saaidah, A.; Oraiqat, M. A new phishing-website detection framework using ensemble classification and clustering. *Int. J. Data Netw. Sci.* **2023**, *7*, 857–864. [CrossRef]
50. Gualberto, E.S.; De Sousa, R.T.; Vieira, T.P.D.B.; Da Costa, J.P.C.L.; Duque, C.G. The answer is in the text: Multi-stage methods for phishing detection based on feature engineering. *IEEE Access* **2020**, *8*, 223529–223547. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.