Phishing Message Detection Based on Keyword Matching

Keng-Theen Tham
Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

Kok-Why Ng
Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

Su-Cheng Haw
Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

Abstract: This paper proposes to use the Naïve Bayes-based algorithm for phishing detection, specifically in spam emails. The paper compares probability-based and frequency-based approaches and investigates the impact of imbalanced datasets and the use of stemming as a natural language processing (NLP) technique. Results show that both algorithms perform similarly in spam detection, with the choice between them depending on factors such as efficiency and scalability. Accuracy is influenced by the dataset configuration and stemming. Imbalanced datasets lead to higher accuracy in detecting emails in the majority class, while they struggle to classify minority-class emails. In contrast, balanced datasets yield overall high accuracy for both spam and ham email identification. This study reveals that stemming has a minor impact on algorithm performance, occasionally decreasing in accuracy due to word grouping. Balancing the dataset is crucial for improving algorithm performance and achieving accurate spam email detection. Hence, both probability-based and frequency-based Naïve Bayes algorithms are effective for phishing detection using balanced datasets. The frequency-based approach, with a balanced dataset and stemming, achieves a balanced performance between recall and precision, while the probability-based method with a balanced dataset and no stemming prioritises overall accuracy.

Keywords: keyword matching, phishing detection, Naïve Bayes, natural language processing, stemming

Introduction

In today's digital age, communication is predominantly conducted through email and Short Message Service (SMS). These channels serve as essential means for various purposes, including business transactions involving substantial amounts of money and important notifications from subscribed companies or government entities. However, this convenience also attracts malicious individuals who attempt to deceive others by impersonating reputable entities, a practice commonly known as phishing. According to Cveticanin (2023), a study from Verizon shows that one-third of the data breaches in 2018 were caused by phishing attacks. An article from Comparitech stated that financial services are the biggest targets for phishing attacks, based on statistics provided by the Anti-Phishing Working Group (AWPG) (Cook, 2023). In the statistics provided by the AWPG, Software as a Service and webmail were the second most targeted in phishing attacks.

Desolda *et al.* (2022) stated that factors contributing to successful phishing attacks include a lack of knowledge, distraction, fatigue, pressure and a lack of awareness. Jari (2022) mentioned that the human factors leading to phishing victimisation are reciprocation, consistency and commitment, social proof, liking, authority and scarcity, where these terms are described as follows.

- Reciprocation
  Attackers may send phishing emails that appear to offer something valuable or urgent, such as free items or financial opportunities. This is to encourage recipients to click a malicious link or provide personal information.
- Consistency and commitment
  Attackers may craft emails that mimic legitimate organizations or services, relying on the recipients' previous commitments to those entities. This attack can be done by sending fake account verification or password reset emails.
- Social proof
  Phishing emails may contain fake testimonials or user reviews, giving the impression that others have already followed suit. This allows the recipients to be more comfortable taking the desired action, even if the social proof is fabricated.
- Liking
  Attackers may pose as individuals or entities that the recipients may like or trust, such as friends, colleagues or family members. Through this action the attacker can increase the likelihood that the recipients will engage with their malicious content.
- Authority
  Attackers may impersonate authoritative figures or trusted institutions, such as banks or government agencies. This misrepresentation of authority can convince the recipient that they are interacting with a credible source and will be more likely to comply with the attackers' requests.
- Scarcity
  Phishing emails may create a sense of urgency or scarcity to encourage recipients into

taking immediate action, such as claiming that an account will be suspended or an opportunity will expire soon, pushing recipients to act hastily without thinking.

This is also supported by Frauenstein & Flowerday's (2020) study, wherein these factors are also known as the six key principles of persuasion or the six principles of influence. Lin *et al.* (2019) found that older women were the most susceptible to phishing emails, compared to other demographics, and younger users were less susceptible over time, while older users' susceptibility remained the same over the study period. They also concluded that weapons of influence work differently on distinct groups of people, such as younger generations are susceptible to scarcity while older generations are susceptible to reciprocation. Lastly, they proved that the current security training and warning solutions are not suitable for most users. Since different demographics are susceptible to different types of weapons of influence, the current 'one-size-fits-all' security training and warning solutions are less effective.

As most communication nowadays is online, there is a huge increase in spam messages received by users (Tay, 2023). These messages are a waste of time and can cause potential security risks, especially in the case of phishing messages. Although there are built-in spam detection methods in some messaging platforms, not all platforms receive the same treatment. This will lead to users being victims of phishing messages. Therefore, this paper proposes to use a Naïve Bayes-based spam detection method to detect phishing and spam messages received by users. The existing spam detection approaches that solely rely on word frequency may fail to capture the underlying probabilities associated with words occurring in spam emails. This limitation diminishes the overall accuracy of spam detection systems and leaves room for improvement. Therefore, this research also aims to compare the performance of the Naive Bayes classifier using probability-based features against the traditional frequency-based approach in terms of accuracy. Stemming is a technique commonly used in natural language processing (NLP) to reduce words to their base or root form, which can aid in spam detection by capturing the essence of the words without considering their specific variations. However, it remains unclear whether incorporating stemming in spam detection algorithms significantly improves the accuracy of detection compared to approaches that do not use stemming. Hence, one of the objectives of this paper is to compare the accuracy of spam detection methods in different conditions, such as in imbalanced datasets and NLP techniques.

Literature Review

For detecting phishing attacks, Adebowale *et al.* (2019) proposed a method using related features of images, frames and text of both genuine and fraudulent websites to detect a website's legitimacy. They found that although many phishing websites make their websites look similar to the original, there are still many distinctive features that can differentiate the two, such as spelling errors, long URL addresses and image alterations. The features used by their methods include page ranking of the website, the length of the website's URL, identifying abnormal URLs and if the website is using any URL-shortening services.

In addition, features such as if the website submits information to any personal email and the layout similarity of the website are critical in detecting if a website is illegitimate. Combined with the Adaptive Neuro-Fuzzy Inference System model and the Sugeno fuzzy model, the approach yielded an accuracy of 98.3%, showing success as an integrated solution for detecting web phishing. According to the authors, their approach is based on a scheme proposed by Aburrous *et al.* (2010) and a similar scheme proposed by Barraclough & Sexton (2015), both of which suggested using fuzzy techniques to detect phishing.

Aljofey *et al.* (2022) proposed a similar approach that uses machine learning and hybrid feature sets, such as the URL character sequence, hyperlink features and term frequency-inverse document frequency (TF-IDF) character-level features from the plaintext and noisy part of the web page's Hypertext Markup Language (HTML). Classification algorithms, such as eXtreme Gradient Boosting, random forest (RF), logistic regression and AdaBoost classifiers, were used to train the proposed approach. It managed to meet the requirements of real-time detection, third-party independence and high detection efficiency. However, the approach has limitations, such as its dependence on the English language and its inability to detect attached malware because it is incapable of reading and processing external files from a website. Nonetheless, the proposed approach achieved great results, reaching 96.76% accuracy, 98.28% precision and an F1-score of 96.38%.

Table 1 lists the pros and cons of the existing phishing methods.

Table 1. Literature on phishing detection

Method	Author	Pros	Cons
Phishing detection based on hyperlinks using the K-nearest neighbour algorithm	Nurul & Isredza (2021)	Achieved accuracy of 97.80% and 99.60% with 2 datasets consisting of 500 URLs, respectively.	Can only detect phishing attempts related to COVID-19. Can only detect emails that contain a URL. Currently not executed on online websites.
Phishing detection using deep learning technique	Mughaid *et al.* (2022)	Datasets contain distinctive features; thus, the result is more accurate. Achieved 97.7% accuracy using the neural network algorithm.	Feature selection needs more improvement. Lack of an automated tool to extract new features from new raw emails.
Phishing detection through a Bayesian algorithm	Baykara & Gurel (2018)	Manually add spam keywords and URLs. Contains Graphic User Interface.	Program able to directly connect to Gmail inbox; therefore, sensitive emails may be read. Only works with Gmail.
Phishing detection through machine learning approach	Mohamed *et al.* (2022)	Achieved accuracy of 95.18% using neural network	Only works for emails containing URLs.

Method	Author	Pros	Cons
Phishing detection in Short Message Service (SMS) based on multiple correlation algorithms	Sonowal (2020)	Achieved accuracy of 98.40% via the Kendall ranking algorithm with AdaBoost classifiers. Lessened the number of features by 61.53%.	Time consuming.
Phishing detection through multi-layer perceptron (MLP) and random forest (RF) classification algorithms	Dalia *et al.* (2021)	Achieved accuracy of 99.46% using MLP and RF.	
SMS spam detection using content-based features and averaged neural network	Sheikhi *et al.* (2020)	Achieved accuracy of 98.8% using an averaged neural network with selected features.	Needs more records for better classification accuracy. Lack of standard sizable dataset.
Spam detection in SMS using machine learning through text mining	Julis & Alagesan (2020)	Achieved accuracy of 98% using support vector machine. Naïve Bayes has the fastest prediction time.	Cannot add additional filtering techniques or change current aspects.
SMS spam detection using the term frequency-inverse document frequency (TF-IDF) and RF algorithms.	Amir *et al.* (2019)	Accurately classifies 97.50% and achieves 98% precision using TF-IDF with RF.	Performance is lacking. Trained data are not up to standard.

Proposed Methodology

In this paper, two approaches for spam detection in Naïve Bayes were built and compared: probability-based and frequency-based approaches. Both methods were tested with and without stemming to determine if stemming affects the accuracy of Naïve Bayes in detecting spam.

Raw data

In this paper, spam and ham emails from the Spam Assassin Corpus were collected for training and testing. In this case, spam emails refer to emails that are unwanted, such as advertisements or scams, while ham emails refer to emails that are intended or are safe and legitimate emails. A total of 4,638 spam and ham emails were used for training. Based on these emails, two datasets were created. The first dataset consisted of all the collected emails, which included 1,989 spam emails and 2,649 ham emails. The second dataset consisted of 501 spam emails and 501 ham emails. The creation of the second dataset was to determine if imbalanced data would affect the accuracy of spam detection. Here, 598 emails from the Corpus were used for testing: 299 spam emails and 299 ham emails. A summary of the datasets is shown in Table 2.

Table 2. Dataset summary

Dataset	Spam Email	Ham Email	Total
Imbalanced, stemmed + not stemmed	1,989	2,649	4,638
Balanced, stemmed + not stemmed	501	501	1,002
Testing	299	299	598

Processing data

To process the data in JavaScript Object Notation (JSON) format, each email was first parsed to extract the actual message from the email. After parsing the email, the message underwent a basic NLP process, which included:

- Word tokenisation
- Stop word removal
- Remove empty tokens
- Remove duplicate words
- Remove numbers and single-letter words.

When creating the dataset for testing the stemming approach, the message underwent an additional process—stemming. For this process, the Porter–Stemmer algorithm was applied.

After NLP, the message, which was now an array of words, was saved in JSON format, which includes information such as the word, its spam frequency, ham frequency, spam probability and ham probability. The spam and ham probabilities were calculated using the following formulas:

$$\boldsymbol{Spam\ Probability} = \frac{\boldsymbol{Spam\ Frequency}}{\boldsymbol{Spam\ Frequency+Ham\ Frequency}} \quad (1)$$

$$\boldsymbol{Ham\ Probability} = \frac{\boldsymbol{Spam\ Frequency}}{\boldsymbol{Spam\ Frequency+Ham\ Frequency}} \quad (2)$$

Classification algorithm

Two classification algorithms were created in this paper: Naïve Bayes with frequency-based features and with probability-based features. Before classifying an email, it went through the same basic NLP process as outlined above.

Naïve Bayes with frequency-based features

In Naïve Bayes with frequency-based features, the probabilities of an email being ham or spam without considering its features will first be calculated. These probabilities are known as prior ham and prior spam and their formulas are as follows:

$$\boldsymbol{Prior\ Spam\ Probability} = \frac{\boldsymbol{Total\ Number\ of\ Spam\ Emails}}{\boldsymbol{Total\ Number\ of\ Emails}} \quad (3)$$

$$\boldsymbol{Prior\ Ham\ Probability} = \frac{\boldsymbol{Total\ Number\ of\ Ham\ Emails}}{\boldsymbol{Total\ Number\ of\ Emails}} \quad (4)$$

To verify the probability that an email is spam or ham, the spam and ham scores are calculated based on the following formulas:

$$\boldsymbol{Spam\ Score = Prior\ Spam\ Probability * Feature1\ Spam\ Probability * \dots} \tag{5}$$

$$\boldsymbol{Ham\ Score = Prior\ Ham\ Probability * Feature1\ Ham\ Probability * \dots} \tag{6}$$

When a decimal number is repeatedly multiplied by other decimal numbers, the value gradually decreases towards 0. This phenomenon is known as 'decimal underflow' or 'floating-point overflow', and it is solved by using the logarithmic exponential technique. The basic idea of this technique is to perform calculations in the logarithmic domain instead of the original decimal domain. This is because by working with logarithms, which are additive, instead of exponentials, which are multiplicative, the calculation will be more stable and accurate, especially when dealing with extreme values. After performing the necessary operations in the logarithmic domain, the result is exponentiated back to the original domain. However, due to the nature of the spam and ham probability being too small, a slight modification must be applied to the technique. Instead of performing the exponential function first and applying the logarithmic function later, the logarithmic function must be applied first, as performing the exponential function on the original feature's probability will not result in considerable change.

Using the above technique, the new formulas would be:

$$\boldsymbol{Spam\ Score = Exp(Log(Prior\ Spam\ Probability) + Log(Feature1\ SP) * \dots)} \tag{7}$$

$$\boldsymbol{Ham\ Score = Exp(Log(Prior\ Ham\ Probability) + Log(Feature1\ HP) * \dots)} \tag{8}$$

***SP** = Spam probability; ***HP** = Ham probability

To calculate each feature's spam and ham probabilities, the following formulas are used:

$$\boldsymbol{Feature\ Spam\ Probability = \frac{Feature\ Spam\ Frequency}{Total\ Number\ of\ Spam\ Frequency}} \tag{9}$$

$$\boldsymbol{Feature\ Ham\ Probability = \frac{Feature\ Ham\ Frequency}{Total\ Number\ of\ Ham\ Frequency}} \tag{10}$$

Using the above formulas, if a feature has never been registered as spam or ham, the feature's spam or ham probability will be 0, which is also known as the zero probabilities problem. Therefore, when calculating the score, the score will be undefined, as log(0) returns undefined. In this case, Laplace smoothing is employed to solve this problem. Laplace smoothing adds a constant value, usually 1, to the observed frequencies of each feature. Here, the probability estimate will never be 0, and it also evenly distributes the probability mass across all features. Hence, the new formulas would be:

$$\boldsymbol{Feature\ Spam\ Probability = \frac{Feature\ Spam\ Frequency + 1}{Total\ Number\ of\ Spam\ Frequency + Total\ Number\ of\ Features}} \tag{11}$$

$$\boldsymbol{Feature\ Ham\ Probability} = \frac{\boldsymbol{Feature\ Ham\ Frequency + 1}}{\boldsymbol{Total\ Number\ of\ Ham\ Frequency + Total\ Number\ of\ Features}} \quad (12)$$

To calculate the spam and ham probabilities of an email, the formulas are:

$$\boldsymbol{Spam\ Probability} = \frac{\boldsymbol{Spam\ Score}}{\boldsymbol{Spam\ Score + Ham\ Score}} * \mathbf{100\%} \quad (13)$$

$$\boldsymbol{Ham\ Probability} = \frac{\boldsymbol{Ham\ Score}}{\boldsymbol{Spam\ Score + Ham\ Score}} * \mathbf{100\%} \quad (14)$$

Naïve Bayes with probability-based features

In Naïve Bayes with probability-based features, the probabilities of an email being ham or spam without considering its features will first be calculated. These probabilities are known as prior ham and prior spam and their formulas are as follows:

$$\boldsymbol{Prior\ Spam\ Probability} = \frac{\boldsymbol{Total\ Number\ of\ Spam\ Emails}}{\boldsymbol{Total\ Number\ of\ Emails}} \quad (15)$$

$$\boldsymbol{Prior\ Ham\ Probability} = \frac{\boldsymbol{Total\ Number\ of\ Ham\ Emails}}{\boldsymbol{Total\ Number\ of\ Emails}} \quad (16)$$

To calculate the probability that an email is spam or ham, the spam and ham scores will first be computed using the formulas stated previously, with slight modification. The modification applied here is that instead of performing the logarithmic operation first, the exponential operation is performed first. This is because, here, each feature's probability value is significantly higher compared to the values calculated in the frequency-based algorithm. Therefore, the original logarithm exponential technique can be used in this case:

$$\boldsymbol{Spam\ Score} = \boldsymbol{Log}(\boldsymbol{Exp}(\boldsymbol{Prior\ Spam\ Probability}) + \boldsymbol{Exp}(\boldsymbol{Feature1\ SP}) * \ldots) \quad (17)$$

$$\boldsymbol{Ham\ Score} = \boldsymbol{Log}(\boldsymbol{Exp}(\boldsymbol{Prior\ Ham\ Probability}) + \boldsymbol{Exp}(\boldsymbol{Feature1\ HP}) * \ldots) \quad (18)$$

***SP** = Spam probability; ***HP** = Ham probability

To calculate each feature's spam and ham probabilities, the following formulas are used:

$$\boldsymbol{Feature\ Spam\ Probability} = \frac{\boldsymbol{Feature\ Spam\ Frequency}}{\boldsymbol{Feature\ Spam\ Frequency + Feature\ Ham\ Frequency}} \quad (19)$$

$$\boldsymbol{Feature\ Ham\ Probability} = \frac{\boldsymbol{Feature\ Ham\ Frequency}}{\boldsymbol{Feature\ Spam\ Frequency + Feature\ Ham\ Frequency}} \quad (20)$$

Using the above formulas, if a feature has never been registered as spam or ham, the feature's spam or ham frequency will be 0, which will also lead to the zero probabilities problem. Therefore, when calculating the score, the score will be undefined, as log(0) returns undefined. In this case, Laplace smoothing is employed to solve this problem, as above. Hence, if the spam or ham probability is 0, the probability will be a small value that will not greatly affect the result, which was 0.00001 here.

To calculate the spam and ham probabilities of an email, the following formulas are used:

$$\boldsymbol{Spam\ Probability} = \frac{\boldsymbol{Spam\ Score}}{\boldsymbol{Spam\ Score + Ham\ Score}} * \mathbf{100\%} \tag{21}$$

$$\boldsymbol{Ham\ Probability} = \frac{\boldsymbol{Ham\ Score}}{\boldsymbol{Spam\ Score + Ham\ Score}} * \mathbf{100\%} \tag{22}$$

Results, Analysis, and Discussions

To test the accuracy of the algorithms, a testing dataset that contains 299 spam emails and 299 ham emails from the Spam Assassin Corpus was used. The testing was performed on the probability-based and the frequency-based algorithms. Both algorithms were evaluated under different conditions: whether stemming was applied, and whether a balanced dataset was used. For each combination of the algorithm, the dataset and the use of stemming, the number of correctly and incorrectly classified spam and ham emails was recorded. These results were then used to calculate the accuracy of each algorithm under specific conditions. The results of the testing are shown in Table 3, where the accuracy for each situation is recorded. The accuracy is shown as the percentage of correctly classified emails out of the total number of emails in the spam and ham categories.

Based on the results for the frequency-based approach, a comparable performance to the probability-based algorithm was seen across different dataset scenarios. Both the stemmed and non-stemmed versions of the algorithm obtained high spam accuracies of 96.66% and 95.99%, respectively, in the unbalanced dataset. This suggests that a significant number of spam emails were appropriately identified by the algorithm. The ham accuracy numbers were still high, showing that ham emails were also correctly classified. The frequency-based method worked remarkably well in the balanced dataset case, similar to the probability-based approach. With high spam accuracies of 96.99% and 97.32%, respectively, both the stemmed and non-stemmed versions successfully classified the spam emails. The ham accuracies remained constant at 100%, indicating that ham emails can be correctly identified.

Table 3. The accuracy values for the probability-based and frequency-based algorithms under different datasets, stemming and spam/ham emails

Algorithm	Dataset Used	Stemming	Spam/Ham	Correct	Wrong	Accuracy
Probability-based	Imbalanced	Yes	Spam	182	117	60.87%
Probability-based	Imbalanced	Yes	Ham	299	0	100.00%
Probability-based	Imbalanced	No	Spam	227	72	75.92%
Probability-based	Imbalanced	No	Ham	299	0	100.00%
Probability-based	Balanced	Yes	Spam	295	4	98.66%
Probability-based	Balanced	Yes	Ham	296	3	99.00%

Algorithm	Dataset Used	Stemming	Spam/Ham	Correct	Wrong	Accuracy
Probability-based	Balanced	No	Spam	295	4	98.66%
Probability-based	Balanced	No	Ham	298	1	99.67%
Frequency-based	Imbalanced	Yes	Spam	289	10	96.66%
Frequency-based	Imbalanced	Yes	Ham	293	6	97.99%
Frequency-based	Imbalanced	No	Spam	287	12	95.99%
Frequency-based	Imbalanced	No	Ham	292	7	97.66%
Frequency-based	Balanced	Yes	Spam	290	9	96.99%
Frequency-based	Balanced	Yes	Ham	299	0	100.00%
Frequency-based	Balanced	No	Spam	291	8	97.32%
Frequency-based	Balanced	No	Ham	299	0	100.00%

Based on the results for the probability-based approach, the accuracy for identifying spam emails was low, at 60.87%, when working with an unbalanced dataset and using stemming. This demonstrates that the algorithm had trouble correctly classifying spam emails. However, the algorithm successfully identified ham emails, with a ham accuracy of 100%. The approach somewhat improved in spam detection, reaching 75.92%, when stemming was not performed on the unbalanced dataset. Regardless of whether stemming was employed, the algorithm exhibited great accuracy in recognising spam and ham emails when utilising a balanced dataset. The algorithm was capable of 98.83% and 99.17% overall accuracy in stemmed and non-stemmed scenarios, respectively, based on the results.

To assess each algorithm's performance, metrics such as precision, recall, F1-score and accuracy were used (Harikrishnan, 2021). For this assessment, true positives were spam emails that were correctly identified, true negatives were ham emails that were correctly identified, false positives were ham emails that were incorrectly identified as spam emails and false negatives were spam emails that were incorrectly identified as ham emails. Each metric is explained below.

Precision

Precision was used to measure the proportion of correctly classified spam emails out of all emails that were classified as spam. It indicates the reliability of the algorithm's identification of spam emails. Precision is calculated using the following formula:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Positives}} \tag{23}$$

Recall

Recall was used to measure the proportion of correctly classified spam emails out of all spam emails. It indicates the algorithm's ability to identify all spam emails. Recall is calculated using the following formula:

$$\boldsymbol{Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}} \tag{24}$$

F1-Score

The F1-score was used to combine precision and recall into a single value, providing a balanced measure of the model's performance. It is calculated using the following formula:

$$\boldsymbol{F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}} \tag{25}$$

Accuracy

Accuracy measures the overall correctness of the model's prediction. It calculates the proportion of correctly identified emails out of the total number of emails tested. Accuracy is calculated using the following formula:

$$\boldsymbol{Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + True\ Negative + False\ Negatives}} \tag{26}$$

The performance metrics of each method are presented in Table 4.

From Table 4, it can be observed that the accuracy of the algorithms varied depending on the usage of the dataset and stemming. When using an imbalanced dataset, where there was a large number of ham emails compared to spam emails, the algorithms tended to achieve higher accuracy in detecting the ham emails. This is because the accuracy is heavily influenced by the majority class. For example, the word 'transaction' appeared 1,000 and 50 times in 2,000 spam emails and 700 ham emails, respectively. If the word 'transaction' is detected in a new email, the probability will favour spam instead of ham due to the dataset used. Therefore, the algorithm and dataset used will have a high accuracy when predicting the majority class but a poor accuracy when predicting the minority class. Stemming, as a NLP technique, had a minor impact on the performance of the algorithms. In certain scenarios, stemming decreased the accuracy of the algorithms. This may be because words can have different meanings when reduced to their base or root forms. For example, the words 'occupant' and 'occupation' can be stemmed into 'occup', which may cause these two different words to be grouped into a single category.

Table 4. Performance metrics for the probability-based and frequency-based Naïve Bayes algorithms in spam email classification

Algorithm	Dataset	Stemming	True Positive	True Negative	False Positive	False Negative	Precision	Recall	F1-Score	Accuracy (%)
Probability-based	Imbalanced	Yes	182	299	0	117	1	0.608	0.756	80.44
Probability-based	Imbalanced	No	227	299	0	72	1	0.759	0.863	87.96
Probability-based	Balanced	Yes	295	296	3	4	0.99	0.986	0.988	98.83
Probability-based	Balanced	No	295	298	1	4	0.996	0.987	0.992	99.17
Frequency-based	Imbalanced	Yes	289	293	6	10	0.98	0.966	0.973	97.33
Frequency-based	Imbalanced	No	287	292	7	12	0.976	0.96	0.968	96.83
Frequency-based	Balanced	Yes	290	299	0	9	1	0.97	0.985	98.5
Frequency-based	Balanced	No	291	299	0	8	1	0.973	0.986	98.66

Both algorithms achieved a higher accuracy, precision, recall and F1-score when working with balanced datasets compared to imbalanced datasets. Using a balanced dataset helps the algorithms to learn more effectively from both types of emails, which increases the performance. Therefore, it can be concluded that balancing the dataset or collecting a balanced dataset could lead to more accurate spam email detection. Comparing the probability-based and frequency-based algorithms, both approaches achieved high accuracy when using a balanced dataset. This indicates that both algorithms are effective in spam email classification tasks.

The probability-based algorithm was significantly weaker when working with an imbalanced dataset. Analysing the result for the probability-based algorithm using an imbalanced dataset, the algorithm achieved 100% accuracy in classifying ham emails but only 60–75% accuracy when classifying spam emails. This is because the algorithm is more biased towards the majority class. Thus, the algorithm will be more likely to predict that an email is ham, leading to a higher number of true negatives and a higher ham accuracy. This also causes a higher number of false negatives, leading to a lower number of true positives and a lower spam accuracy. However, the imbalanced dataset did not greatly affect the accuracy of the frequency-based algorithm compared to the probability-based algorithm. This is because both algorithms calculate the spam probability of a feature in a slightly different way. The probability-based algorithm calculates the spam and ham probabilities by considering the frequency of both spam and ham emails. The frequency-based algorithm, on the other hand, calculates the spam and ham probabilities based on the frequency within all spam emails in a dataset. Since the probability-based algorithm takes both spam and ham frequencies into account, when there is an imbalance, the dominant presence of ham emails influences the calculation of the spam probability. The frequency-based algorithm relies only on a feature's frequency within all spam emails and, therefore, is less likely to be influenced by the dataset's imbalance.

Two combinations stood out in terms of the F1-score, which reflects the balance between precision and recall. The first combination achieved an F1-score of 0.985 using the frequency-based algorithm with a balanced dataset and stemming. With this combination, spam and ham emails can be distinguished with high accuracy. The second combination achieved a remarkable F1-score of 0.992 using the probability-based algorithm with a balanced dataset and no stemming. This combination shows that both spam and ham emails can be classified with an elevated level of recall and precision. The probability-based algorithm with a balanced dataset and no stemming, however, surpassed the others, obtaining an accuracy of 99.17%, considering overall accuracy as the main factor. The accuracy of the probability-based algorithm using a balanced dataset and stemming was 98.83%, which was the second-highest overall accuracy among all combinations.

The ideal combination will be determined by the needs and priorities of the phishing detection system. For a balanced performance between recall and precision, the frequency-based approach with a balanced dataset and stemming is a great option. On the other hand, if overall accuracy is the main concern, the probability-based method with a balanced dataset and no stemming would be the best option.

Conclusion

The performance of the spam detection algorithms varied depending on the dataset configuration and the presence of stemming. The results indicated that using a balanced dataset led to a higher accuracy, precision, recall and F1-score for both the probability-based and frequency-based approaches. Balancing the dataset proved to be crucial in achieving accurate spam email detection. Stemming had a minor impact on the algorithms' performance, sometimes even decreasing the accuracy due to the grouping of words with different meanings into a single category.

Comparing the two algorithms, both the probability-based and frequency-based approaches demonstrated effectiveness in spam email identification when using a balanced dataset. However, the probability-based algorithm showed weakness when dealing with an imbalanced dataset, with a lower accuracy in classifying spam emails compared to ham emails. The frequency-based algorithm, on the other hand, was less affected by dataset imbalances, as it relied on the frequency of features within all spam emails.

Based on the results, the ideal combination for creating a strong spam detection system would depend on the specific requirements and priorities of the system. If a balanced performance between recall and precision is desired, the frequency-based approach with a balanced dataset and stemming could be chosen. However, if overall accuracy is the main concern, the probability-based method with a balanced dataset and no stemming would be a preferable option.

Acknowledgements

A version of this paper was presented at the 3rd International Conference on Computer, Information Technology and Intelligent Computing (CITIC 2023), held in Malaysia on 26–28 July 2023.

References

Aburrous, M., Hossain, M. A., Dahal, K. & Thabtah, F. (2010). Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Systems with Applications, 37*(12), 7913–7921. https://doi.org/10.1016/j.eswa.2010.04.044

Adebowale, M. A., Lwin, K. T., Sánchez, E. & Hossain, M. A. (2019). Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text. *Expert Systems with Applications, 115*, 300–313. https://doi.org/10.1016/j.eswa.2018.07.067

Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q. & Wang, Y. (2022). An effective detection approach for phishing websites using URL and HTML features. *Scientific Reports, 12*(1). https://doi.org/10.1038/s41598-022-10841-5

Amir Sjarif, N. N., Mohd Azmi, N. F., Chuprat, S., Sarkan, H. M., Yahya, Y. & Sam, S. M. (2019). SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science, 161*, 509–515. https://doi.org/10.1016/j.procs.2019.11.150

Barraclough, P. & Sexton, G. (2015). *Phishing website detection fuzzy system modelling* [Paper presentation]. 2015 Science and Information Conference (SAI). https://doi.org/10.1109/sai.2015.7237323

Baykara, M. & Gurel, Z. Z. (2018). *Detection of phishing attacks* [Paper presentation]. 2018 6th International Symposium on Digital Forensic and Security (ISDFS). https://doi.org/10.1109/isdfs.2018.8355389

Cook, S. (2023, 21 June). *50+ Phishing statistics, facts and trends 2017–2018*. Comparitech. https://www.comparitech.com/blog/vpn-privacy/phishing-statistics-facts/

Cveticanin, N. (2023, 14 July). *Phishing statistics & how to avoid taking the bait*. Dataprot. https://dataprot.net/statistics/phishing-statistics/

Dalia, S. A., Hanan, A. A. A. A. & Ishraq, K. A. (2021). Effective phishing emails detection method. *Turkish Journal of Computer and Mathematics Education, 12*(14), 4898–4904. https://turcomat.org/index.php/turkbilmat/article/view/11456

Desolda, G., Ferro, L. S., Marrella, A., Catarci, T. & Costabile, M. F. (2022). Human factors in phishing attacks: A systematic literature review. *ACM Computing Surveys*, *54*(8), 1–35. https://doi.org/10.1145/3469886

Frauenstein, E. D. & Flowerday, S. (2020). Susceptibility to phishing on social network sites: A personality information processing model. *Computers & Security, 94*, 101862. https://doi.org/10.1016/j.cose.2020.101862

Harikrishnan N B. (2021, 13 December). *Confusion matrix, accuracy, precision, recall, F1 score*. Medium. https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd

Jari, M. (2022). *An overview of phishing victimization: Human factors, training and the role of emotions* [Paper presentation]. 12th International Conference on Computer Science and Information Technology. https://doi.org/10.5121/csit.2022.121319

Julis, M. & Alagesan, S. (2020). Spam detection in SMS using machine learning through text mining. *International Journal of Scientific & Technology Research, 9*. Available at https://www.ijstr.org/final-print/feb2020/Spam-Detection-In-Sms-Using-Machine-Learning-Through-Text-Mining.pdf

Lin, T., Capecci, D. E., Ellis, D. M., Rocha, H. A., Dommaraju, S., Oliveira, D. S. & Ebner, N. C. (2019). Susceptibility to spear-phishing emails. *ACM Transactions on Computer–Human Interaction, 26*(5), 1–28. https://doi.org/10.1145/3336141

Mohamed, G., Visumathi, J., Mahdal, M., Anand, J. & Elangovan, M. (2022). An effective and secure mechanism for phishing attacks using a machine learning approach. *Processes, 10*(7), Article 1356. https://doi.org/10.3390/pr10071356

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*, 3819–3828. https://doi.org/10.1007/s10586-022-03604-4

Nurul, A. A. & Isredza, R. A. H. (2021). COVID-19 phishing detection based on hyperlink using K-nearest neighbor (KNN) algorithm. *Applied Information Technology and Computer Science, 2*(2), 287–301. Available from https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2317

Sheikhi, S., Taghi Kheirabadi, M. & Bazzazi, A. (2020). An effective model for SMS spam detection using content-based features and averaged neural network. *International Journal of Engineering, Transactions B: Applications, 33*(2), 221–228. http://dx.doi.org/10.5829/ije.2020.33.02b.06

Sonowal, G. (2020). Detecting phishing SMS based on multiple correlation algorithms. *SN Computer Science, 1*(6). https://doi.org/10.1007/s42979-020-00377-8

Tay, Y. H., Ooi, S. Y., Pang, Y. H., Gan, Y. H., & Lew, S. L. (2023). Ensuring Privacy and Security on Banking Websites in Malaysia: A Cookies Scanner Solution. *Journal of Informatics and Web Engineering*, 2(2), 153-167. https://doi.org/10.33093/jiwe.2023.2.2.12