Educational Administration: Theory and Practice
2023, 29(4),6326-6332
ISSN: 2148-2403
https://kuey.net/ **Research Article**

A Novel Combating Email Spam and Phishing Classifier Using Intelligent Multinomial Naive Bayes Classification

Zahid Hussain Wani[1], Hilal Ahmad Khanday[2*]

[1,2]University of Kashmir, Srinagar, India

Citation: Zahid Hussain Wani, et.al (2023). A Novel Combating Email Spam and Phishing Classifier Using Intelligent Multinomial Naive Bayes Classification, *Educational Administration: Theory and Practice*, 29(4)01-04
Doi: 10.53555/kuey.v29i4.11435

ARTICLE INFO	ABSTRACT

Since email is still one of the most popular means of communication, it is frequently the subject of phishing and spam attacks, which put people's and organisations' security at grave danger. Conventional detection methods, such signature-based filtering and blacklisting, frequently fall short of spotting dynamic phishing patterns and transient harmful links. This work suggests an automated email spam classification system using the Multinomial Naive Bayes method to overcome these drawbacks. The program successfully differentiates between spam and authentic (ham) emails by using word frequency analysis and textual feature extraction. Multinomial Naive Bayes exhibits dependable results in identifying spam content because of its simplicity, effectiveness, and solid probabilistic base. The suggested method provides a scalable and efficient way to improve email security and reduce cyber threats associated with phishing.

Introduction

With billions of people around the world utilizing email as an effective way of formal communication [1], it is important to keep this mode of communication safe and encrypted [2]. However, as the number of people using email is increasing, so are the people sending out spam emails [3]. With just an accidental click or opening up one spam email, one might be putting their organization, or personal information at security risk [4][5]. Spamming includes unwanted emails being sent to large group of people, however, humans are indeed prone to make errors, hence even with filters to scan out these scam emails, it does not necessarily allow complete feeling of security [6].

Eighty one percent of the time cyber attacks against organizations have occurred due to spam emails [6]. This calls on the idea that we are in desperate need for an efficient way in determining scam emails [7]. Organizations have been using multiple ways to strengthen their email system by using factors like subdomain control which include creation of a different domain specifically for email, educating users and more [8][9]. Additionally, some generalized ways of detecting scam emails include Blacklisting which includes listing those resources who were involved in previous phishing attacks [10], or signature based approach which looks for patterns detected in previous phishing attacks [11]. However both these techniques have their own negatives that makes them unreliable. Blacklisting is not very reliable due to short term phishing links, difficulty maintaining a comprehensive list, and the ease on the scammers end as they could just make a minor change in a URL [12], and they would go unnoticed. Similarly, a signature based approach may flag a legitimate email, relying on old phishing patterns which makes them fail to detect new ones [13]

This calls out for the importance of new effective ways in which phishing attacks can be minimised [14]. The Multinomial Naive Bayes algorithm can be utilized here as it can help with efficient ways in which we can classify scam emails [15][16][17]. It is simple to implement and has higher accuracy than some other spam detection [18]. Hence in this proposed study, we aim to build an email spam classifier utilizing Multinomial Naive Bayes algorithm. Based on the frequency of words in the text, the proposed model will be able to classify an email as spam or ham.

Data Cleaning and Preprocessing

Prior to feature extraction and model training, a number of data cleaning procedures were used to guarantee the dataset's quality and integrity. This allowed for the detection and correction of inaccuracies and inconsistencies in the data. The dataset was initially analyzed using the Pandas library to get a better understanding of the structure of the data.

Subsequently, the dataset was assessed for duplicate and missing values. Pandas was used to check for missing values, and the results showed that there were no missing data entries. Additionally, duplicate messages were found and eliminated to ensure redundancy did not bias the model. This ensured that redundant data did not skew the training process. By removing these discrepancies, the dataset was ready for feature extraction and model training. The Category and Message columns were renamed to Label and Text, respectively, for uniformity and clarity. A new column called Spam was created to hold the numerical values obtained from converting the categorical classifications into values where Ham = 0 and Spam = 1. This change was required to facilitate training machine learning models.

The improved dataset was saved into a new file to complete the cleaning procedure. Prior to model training, the dataset was thoroughly cleaned to ensure data integrity. Initially, the dataset included 5,572 entries. After cleaning, 5,157 entries remained, with 415 records being eliminated due to duplication or inconsistency. This made it possible for subsequent processes, such as feature extraction and modeling, to proceed without requiring extra preprocessing. In order to standardize the dataset and prepare it for efficient analysis and spam categorization, the data cleaning step was essential.

Data preparation is an essential step in preparing textual data for machine learning models [19]. It plays a crucial role in enhancing the reliability and accuracy of the model [20]. Machine Learning models require numerical input, so email text must be converted to a structured numerical representation [21]. The Bag of Words (BoW) model, implemented with CountVectorizer from the scikit-learn module, is one of the most effective techniques for achieving this transformation. Text is transformed into a sparse matrix representation via CountVectorizer, where each row represents a distinct email and each unique word is given a column. The frequency of a certain term in an email is indicated by the value in each cell. This method assures that the textual data is uniform and suitable for training a classification model.

In this study, textual data was transformed into a numerical word matrix using CountVectorizer. The cleansed data was preprocessed with CountVectorizer to convert raw email contents into a format that a machine learning model could interpret. Using this feature extraction method, the dataset is tailored for machine learning algorithms, resulting in high accuracy and reliable classification of emails as spam or legitimate messages.

Exploratory Data Analysis

The dataset consists of email messages labeled as either 'spam' or 'ham.' The primary objective is to analyze the data and build a classifier that can accurately distinguish between spam and ham messages.

4.1 Loading and Preprocessing the Dataset

The dataset was loaded from a CSV and inspected for any missing values. A new column named 'Spam' was created to label messages as spam (`1`) or ham (`0`) based on the 'Category' column.The 'Label' column was converted to numerical values: 'ham' was mapped to `0`, and 'spam' was mapped to `1`.The distribution of spam and ham messages was visualized using a count plot as shown in Figure 1. This visualization helps identify the imbalance between the two classes.Word clouds as shown in Figure 2 were generated for spam and ham messages separately to visualize the most frequently occurring words in each category. The length of each message was calculated, and the distribution of message lengths for spam and ham messages was analyzed. Histogram plots as shown in Figure-3 were used to compare the message length distributions between the two categories.

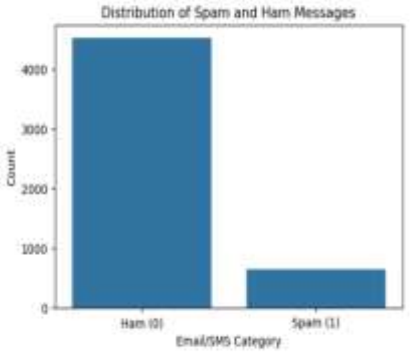
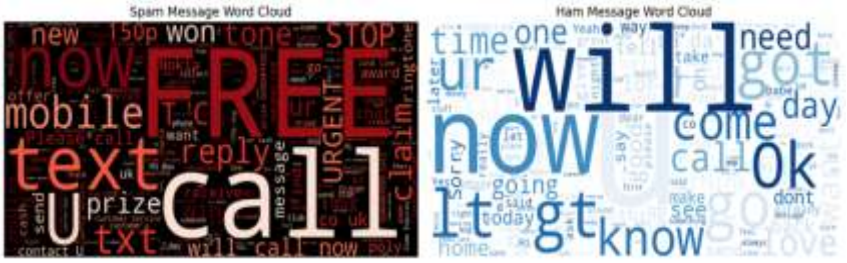


Figure 1 - Count Plot Analysing Class Distribution

Figure 2 – Word



Cloud (Word Frequency Analysis)

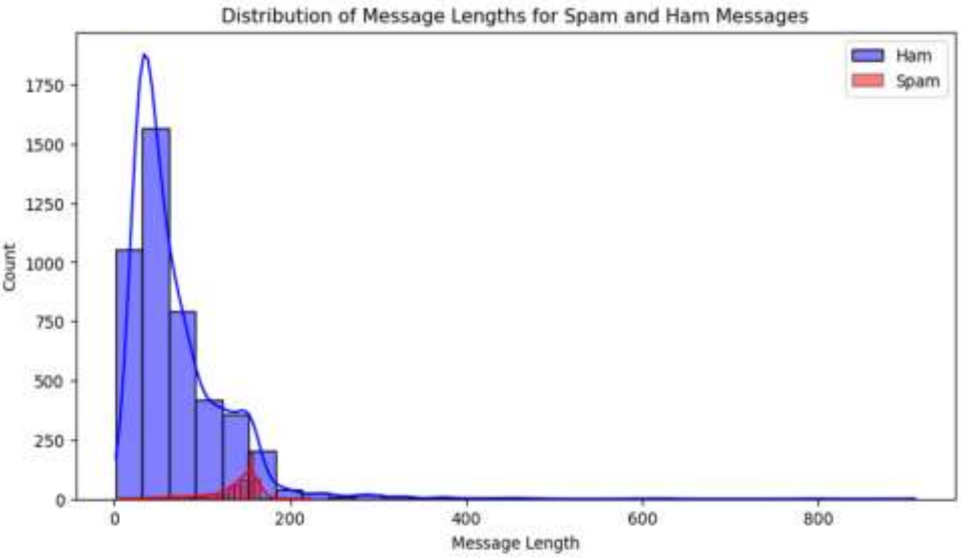


Figure 3 – Histogram Plot showing Message Length Distribution

4.2 Model Selection

Selecting the right model for spam classification involves understanding the dataset, the nature of the features, and the specific requirements of the task. For text classification tasks like spam detection, the following considerations are essential:

- Nature of the Data: Text data often involves a high-dimensional feature space, especially when using techniques like Bag-of-Words or TF-IDF for feature extraction.
- Efficiency and Performance: The chosen model should efficiently handle large datasets and provide accurate classifications.
- Interpretability: For practical applications, it's beneficial to have a model that's not only effective but also interpretable.
- Nature of the Data: Text data often involves a high-dimensional feature space, especially when using techniques like Bag-of-Words or TF-IDF for feature extraction.
- Efficiency and Performance: The chosen model should efficiently handle large datasets and provide accurate classifications.
- Interpretability: For practical applications, it's beneficial to have a model that's not only effective but also interpretable.

4.3 Explored Algorithms

We considered several algorithms for spam classification, each with its strengths and weaknesses:

1- Naive Bayes (Multinomial and Bernoulli)
2- Logistic Regression
3- Support Vector Machines (SVM)
4- Random Forest
5- Deep Learning Models (RNN, CNN)

4.4 Rationale for Choosing Multinomial Naïve Bayes

After evaluating the characteristics of the dataset and the requirements of the classification task, we selected the Multinomial Naïve Bayes algorithm as the final model for the following reasons:

- Simplicity and Interpretability: The Multinomial Naïve Bayes algorithm is straightforward to implement and easy to interpret. It makes strong independence assumptions, which, despite being simplistic, work well in practice for text classification tasks.
- Efficiency: Naïve Bayes algorithms are computationally efficient and require less training time compared to more complex models. This efficiency is particularly advantageous when dealing with large datasets.
- Performance with Text Data: The Multinomial Naïve Bayes algorithm is particularly well-suited for text data, where the features represent the frequency of words in the documents. It has been shown to perform well in spam classification tasks due to its ability to handle high-dimensional feature spaces effectively.
- Handling Word Frequency: - This algorithm treats each feature as the frequency of a word in a document, which aligns well with the nature of the text data used in spam classification. It captures the relationship between word frequency and the likelihood of a message being spam.

- Probabilistic Framework: - The probabilistic framework of Naïve Bayes provides a natural way to combine the evidence from various features. It calculates the posterior probability of each class (spam or ham) given the input features, allowing for a straightforward decision-making process.

4.5 Comparison with Other Models

1. Logistic Regression - While effective, Logistic Regression might require more computational resources and may not perform as well as Naïve Bayes in scenarios with high-dimensional text data.

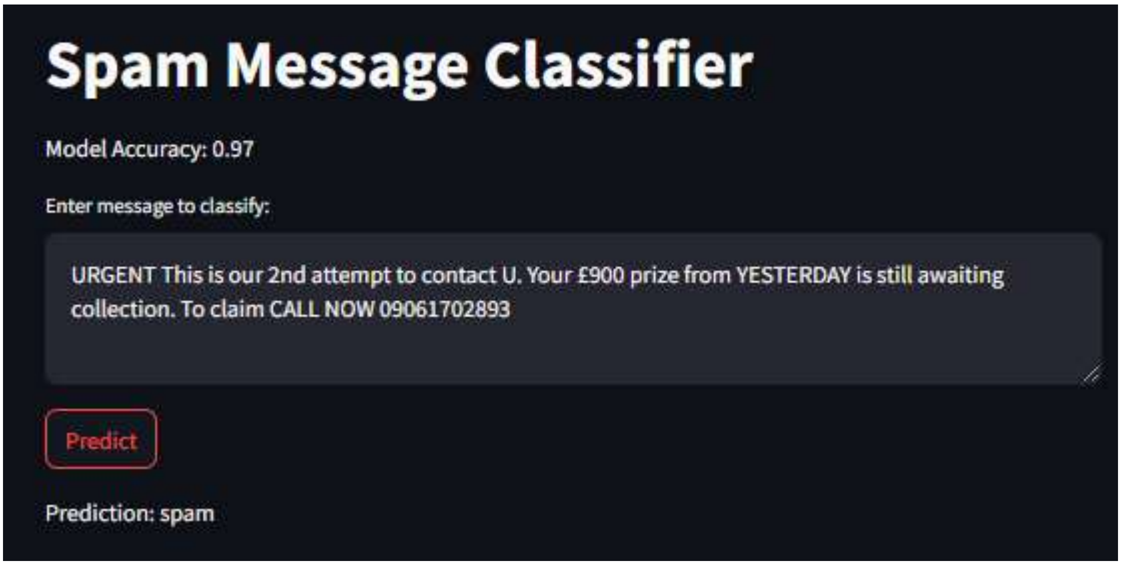
2. Support Vector Machines (SVM) - SVMs are powerful for text classification but can be computationally intensive, especially with large datasets. Tuning hyperparameters for SVMs can also be more challenging.

3. Random Forest - Random Forests provide robustness and handle various data types well. However, they may not perform as efficiently as Naïve Bayes for text data and can be prone to overfitting with high-dimensional features.

4. Deep Learning Models (RNN, CNN)- Deep learning models can achieve high accuracy but require significant computational resources and extensive hyperparameter tuning. They also demand larger training datasets to avoid overfitting and achieve generalization.

Model Analysis

The proposed spam classifier uses the Multinomial Naïve Bayes algorithm to classify emails into spam messages and valid emails. This algorithm depends on the frequency of words in the document and is applied when the data has a multinomial distribution. Using the Multinomial Naïve Bayes algorithm, the model was implemented and trained using the scikit-learn toolkit. The GUI of the implemented model is given below in Figure 4 and Figure 5.



Spam Message Classifier

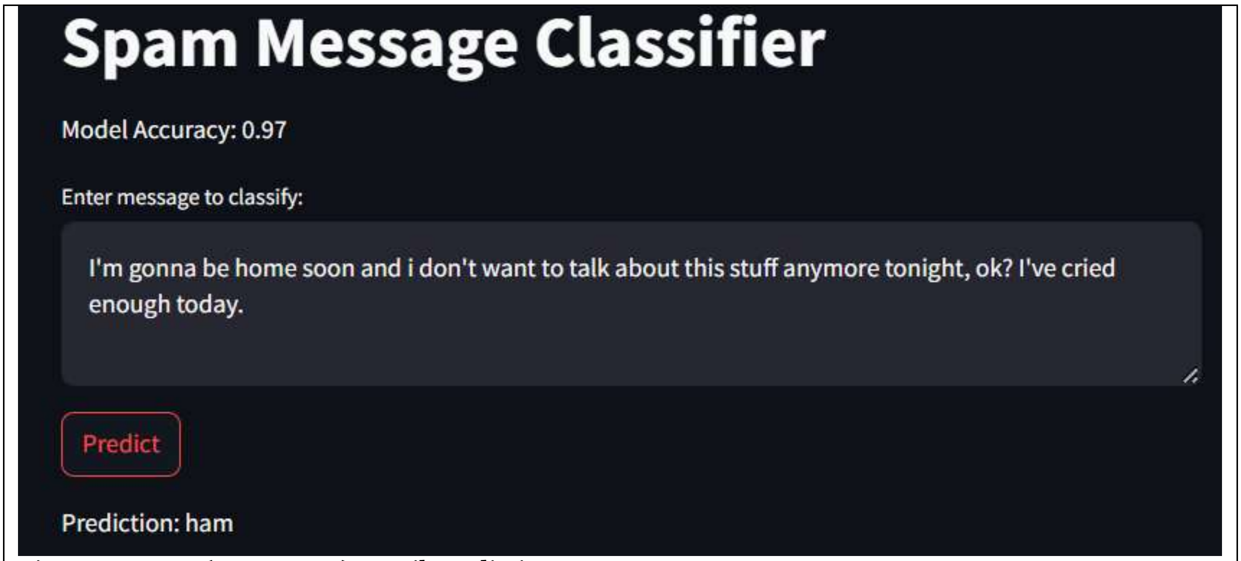
Model Accuracy: 0.97

Enter message to classify:

URGENT This is our 2nd attempt to contact U. Your £900 prize from YESTERDAY is still awaiting collection. To claim CALL NOW 09061702893

Prediction: spam

Figure 4 - Spam Email Prediction


Spam Message Classifier

Model Accuracy: 0.97

Enter message to classify:

I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, ok? I've cried enough today.

Prediction: ham

Figure 5 - Ham (Non-Spam) Email Prediction

4.1 Training and Testing the Model

The dataset was split into 80% training and 20% testing. The model was trained on a corpus of messages labelled as “ham” and “spam”. Using this labelled dataset, the machine learning model learned to categorize spam and legitimate emails. This model was then tested on unseen data to evaluate its effectiveness.

4.2 Performance Evaluation

The effectiveness of the model was evaluated using key performance metrics including accuracy, precision, recall, F1-score and confusion matrix. The model achieved an overall accuracy of 97.19%, indicating high classification performance. The classification report below provides the details about the model's effectiveness.

The model's precision, recall, and F1-score for spam classification were 0.87, 0.91, and 0.89, respectively. This suggests that the model can categorize most spam messages correctly while keeping the false positive rate low. Likewise, the model demonstrated good reliability in identifying legitimate emails with a precision of 0.99, recall of 0.98, and F1-score of 0.98 for the ham classification.

4.3 Confusion Matrix Analysis

The confusion matrix further validates the model's performance. The model was able to correctly classify 886 ham emails while misclassifying 11 as spam. A further 117 spam emails were correctly identified and 18 were mistakenly categorized as ham. This confusion matrix indicates that the number of misclassifications is low, suggesting that the model is efficient in classifying unseen data.

These results indicate that the Multinomial Naïve Bayes model is a robust and efficient choice for email spam classification. However, minor misclassifications still exist, which could be improved through additional feature engineering or ensemble learning techniques.Overall, the Multinomial Naïve Bayes classifier provides a reliable and high-accuracy solution for email spam detection.

Metric	**Precision**	**Recall**	**F1-score**	**Support**
Ham (0)	0.99	0.98	0.98	904
Spam (1)	0.87	0.91	0.89	128
Macro Average	0.93	0.95	0.94	1032
Weighted Average	0.97	0.97	0.97	1032

Table 1- Model Accuracy and Classification report

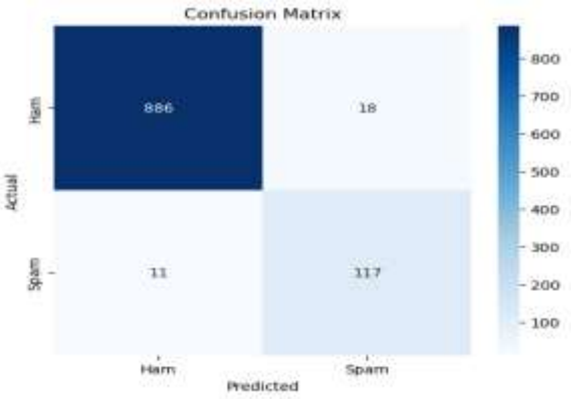


Figure 6 - Confusion Matrix

Conclusion

This implementation of Spam Classifier is a simple yet effective machine learning model. We have used Multinomial Naïve Bayes (MNB) and CountVectorizer to classify messages as either "Spam" or "Ham" (not spam). The model is trained on a labeled dataset of text messages and achieved a reasonable accuracy, as displayed in the Streamlit UI. Our classifier effectively processes user inputs and provides real-time predictions via a web-based interface. Besides this the proposed Naïve Bayes classifier is well-suited for text classification tasks due to its probabilistic approach and efficiency in handling sparse text data,CountVectorizer transforms text into a numerical feature representation, allowing the classifier to understand and process textual input and lastly the Streamlit UI offers an interactive experience where users can input text and receive classification results instantly.

Future Scope

To enhance the classifier's performance and usability, the proposed model can further consider the following improvements in future starting with Feature Engineering & Model Optimization where to use TF-IDF instead of CountVectorizer wherewhile CountVectorizer counts word occurrences, TF-IDF (Term Frequency-Inverse Document Frequency) assigns weights based on importance, potentially improving classification accuracy, incorporate N-grams where Including bigrams and trigrams in text processing can capture contextual meaning better and indeed to use additional NLP Techniques like Implement **stemming, lemmatization, and stopword removal** to refine text preprocessing.

Experimenting with Advanced Models like considering Deep Learning Approaches: Using LSTMs, transformers (e.g., BERT), or CNNs for text classification could provide a more sophisticated approach for spam detection.

Expanding Dataset and Regular Updates simply by using a Larger, More Diverse Dataset which let the model to be trained on a static dataset; adding more real-world messages will enhance its robustness and to continuous model retraining by implement an automated pipeline to update the dataset and retrain the model periodically to adapt to evolving spam tactics.

Deploying the Model using an API instead of embedding everything in Streamlit, the model can be deployed as a REST API using FastAPI or Flask. And the same would allow integration with external applications like email filtering systems, SMS processing services, or chatbot moderation systems.

At final, the current spam classifier serves as a strong foundation for detecting spam messages. By incorporating advanced NLP techniques, improving evaluation metrics, and deploying a more scalable solution, this classifier can be significantly enhanced. The model’s usability can also be extended by deploying it as an API and making the UI more feature-rich.

References

[1] Dürscheid, C., Frehner, C., Herring, S. C., Stein, D., & Virtanen, T. (2013). Email communication. Handbooks of pragmatics [HOPS], (9), 35-54.

[2] Orman, H. (2015). Encrypted Email: The History and Technology of Message Privacy. Springer International Publishing.

[3] Rao, J. M., & Reiley, D. H. (2012). The economics of spam. Journal of Economic Perspectives, 26(3), 87-110.

[4] Silic, M., & Back, A. (2016). The dark side of social networking sites: Understanding phishing risks. Computers in Human Behavior, 60, 35-43.

[5] Krebs, B. (2014). Spam nation: The inside story of organized cybercrime-from global epidemic to your front door. Sourcebooks, Inc..

[6] Rayan, A. (2022).Analysis of email spam detection using novel machine learning-based hybrid bragging technique. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2022/2500772

[7] Wash, R. (2020). How experts detect phishing scam emails. Proceedings of the ACM on Human-Computer Interaction, 4(CSCW2), 1-28.

[8] Turunen, M. (2021). State of email security implementations in Finnish municipalities and joint municipal authorities in 2021: research on current DNS implementations and organizations' publicly available information.

[9] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. Ieee Access, 7, 168261-168295.

[10] Gupta, B. B., Arachchilage, N. A., & Psannis, K. E. (2018). Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems, 67(2), 247-267.

[11] Varshney, G., Misra, M., & Atrey, P. K. (2016). A survey and classification of web phishing detection schemes. Security and Communication Networks, 9(18), 6266-6284.

[12] Bell, S., & Komisarczuk, P. (2020, February). An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank. In Proceedings of the Australasian Computer Science Week Multiconference (pp. 1-11).

[13] Mayer, P., Poddebniak, D., Fischer, K., Brinkmann, M., Somorovsky, J., Sasse, A., ... & Volkamer, M. (2022). " I {don’t} know why I check this..."-Investigating Expert Users' Strategies to Detect Email Signature Spoofing Attacks. In Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022) (pp. 77-96).

[14] Jensen, M. L., Dinger, M., Wright, R. T., & Thatcher, J. B. (2017). Training to mitigate phishing attacks using mindfulness techniques. Journal of Management Information Systems, 34(2), 597-626.

[15] Bermejo, P., Gámez, J. A., & Puerta, J. M. (2011). Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. Expert Systems with Applications, 38(3), 2072-2080.

[16] Chavez, A. (2021). TF-IDF classification based Multinomial Naïve Bayes model for spam filtering (Doctoral dissertation, Dublin, National College of Ireland).

[17] Octaviani, N. L., Rachmawanto, E. H., Sari, C. A., & De Rosal, I. M. S. (2020, September). Comparison of multinomial naïve bayes classifier, support vector machine, and recurrent neural network to classify email spams. In 2020 International seminar on application for technology of information and communication (iSemantic) (pp. 17-21). IEEE.

[18] Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), 23.

[19] Barberio, A. (2022). Large language models in data preparation: opportunities and challenges.

[20] Spasic, I., & Nenadic, G. (2020). Clinical text data in machine learning: systematic review. JMIR medical informatics, 8(3), e17984.

[21] Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. Computers & Security, 110, 102414.