International Journal of Basic and Applied Science 13 (3) (2024) 137-145

Published by: IOCSCIENCE

International Journal of Basic and Applied Science

Journal homepage: www.ijobas.pelnus.ac.id

Performance comparison of naive bayes and support vector machine algorithms in spambot classification in emails

Jonson Manurung[1], Hondor Saragih[2]

[1,2] Informatika, Universitas Pertahanan Republik Indonesia, Bogor, Indonesia

Article Info

Article history:

Received Oct 9, 2024
Revised Nov 20, 2024
Accepted Dec 25, 2024

Keywords:

Classification;
Naive Bayes;
Support Vector Machine;
Spambot;
Email Spam.

ABSTRACT

In the rapidly expanding digital era, email spam poses a significant threat to user productivity and information security. This study aims to comparatively analyze the effectiveness of the Naive Bayes and Support Vector Machine (SVM) algorithms with a radial basis function (RBF) kernel in detecting spambots in emails. The research methodology involves collecting a large-scale email dataset from SpamAssassin, applying both classification algorithms, and evaluating their performance using accuracy, precision, recall, and F1-score metrics. The experimental results indicate that the SVM with an RBF kernel outperforms the Gaussian Naive Bayes model, achieving superior performance across all evaluation metrics. These findings provide valuable insights into the development of more accurate and efficient spam detection systems and emphasize the importance of selecting appropriate machine learning algorithms for complex data classification tasks. Future research could explore the integration of deep learning techniques or hybrid models to enhance detection accuracy and adaptability in evolving spam patterns.

Corresponding Author:

Jonson Manurung,
Program Studi Informatika,
Universitas Pertahanan Republik Indonesia,
Kawasan IPSC Sentul, Sukahati, Kec. Citeureup, Kabupaten Bogor, Jawa Barat 16810, Indonesia.
Email: jhonson.geo@gmail.com

1. INTRODUCTION

In recent decades, email has become one of the primary communication tools in both business and personal settings [1]. However, along with the increased use of email comes serious threats to its security and efficiency, one of which is spambots [2][3][4]. Spambots, software designed to send spam emails en masse, have caused great disruption to the digital communication infrastructure [5]. Not only do spam emails decrease user productivity, but they can also pose security risks, such as the theft of personal data and the spread of malware [6][7]. According to a report from Cybersecurity Ventures (2023), more than 45% of the total emails circulating worldwide are spam emails, with spambots as one of the main contributors. To address this problem, various classification techniques have been developed to distinguish genuine emails from spam [5][8][9]. One of the most commonly used methods is the Naive Bayes and Support Vector Machine (SVM) algorithms [10][11]. These algorithms have proven their ability to process large volumes of text data and automatically filter emails that are indicated as spam. However, with the evolving attack patterns of spambots, it is important to assess the effectiveness and efficiency of these algorithms in handling the growing variety and complexity of email data. The right approach to spam classification is crucial for maintaining network performance and preserving user privacy. Therefore, it is important to comprehensively evaluate the performance of various classification algorithms, especially in the context of spambot detection. This research aims to fill this gap by comparing two commonly used algorithms, Naive Bayes and SVM, in email spambot classification, so as to provide more targeted guidance in choosing a more efficient and accurate detection method [12][13].

While Naive Bayes and Support Vector Machine (SVM) algorithms have been widely used in email spam classification, a major challenge faced today is the increasing sophistication of spambot techniques that are capable of tricking conventional detection systems [14] [15]. Modern spambots not only generate large volumes of spam, but are also capable of manipulating the content to resemble genuine emails, making them difficult to distinguish by standard detection methods. As a result, classification systems often face accuracy issues in identifying spam emails without affecting legitimate emails (false positives) or failing to detect more complex spam (false negatives). Furthermore, there is no clear consensus on which algorithm is superior in handling different types of spam, especially in the context of large data volumes and increasingly diverse content [16][17]. The research conducted by Ahmed, Naeem et al, [18] tends to focus on the application of one of these algorithms individually, without directly evaluating the performance comparison between Naive Bayes and SVM in the face of variations in content and characteristics of modern spambots. This gap is crucial because the right choice of algorithm can have a significant impact on the efficiency and accuracy of spam detection systems in dynamic email environments. Therefore, this research aims to answer an important question: To what extent are Naive Bayes and SVM able to handle spambot classification in emails, especially in the context of large and complex datasets? Does either of these two algorithms show consistent superiority in terms of accuracy and computational efficiency? These questions form the basis of this comparative study, which aims to provide further guidance in selecting the optimal algorithm for spambot detection.

Previous research has explored the use of Naive Bayes and Support Vector Machine (SVM) algorithms in spam email classification, with results showing mixed effectiveness. Naive Bayes, as a probabilistic-based algorithm, is known for its simplicity and computational speed in handling large text data, including spam emails. The study by Zhao, Chensu, et al [3] found that Naive Bayes is able to provide fairly accurate results in an environment where text data is balanced and spam patterns are predictable. However, the study also showed that this algorithm is often less effective in dealing with more dynamic and complex variations in spam content, resulting in a higher rate of false positives. On the other hand, Support Vector Machine (SVM) has been recognized as a more sophisticated algorithm in processing unbalanced data and complex features. The study by Ala'M, et al [19] highlighted that SVM is able to provide higher accuracy in spam classification due to its ability to build stronger decision margins, especially when dealing with high-dimensional data. However, the drawback of SVM lies in its longer computation time and its dependence on parameters that require optimal tuning to achieve the best performance. Although these two algorithms have been studied extensively, several gaps still exist in the literature. Previous studies rarely comprehensively compare the performance of these two algorithms on large and complex spambot datasets, especially in the context of increasingly unpredictable variations in spam content. Therefore, there is an urgent need to conduct research that evaluates a direct comparison between Naive Bayes and SVM in handling spambot classification, focusing on the aspects of accuracy, precision, and computational efficiency. In addition, further development could include the integration of a hybrid approach that combines the advantages of both algorithms to improve spam detection performance in dynamic email environments.

This research aims to conduct a comparative analysis of the performance of two commonly used classification algorithms in email spam detection, namely Naive Bayes and Support Vector Machine (SVM), in the context of spambot classification [20]. Specifically, this research aims to evaluate and compare the two algorithms in terms of accuracy, precision, and computational efficiency when applied to a large and complex email dataset containing a wide variety of modern spam content [21]. This research is expected to provide greater insight into the advantages and limitations of each algorithm in the face of increasing challenges in spambot detection, as well as determine the more optimal method to improve the accuracy and effectiveness of spam classification systems. In addition, this research also aims to identify the conditions under which one algorithm is superior to the other, both in terms of classification performance and computational load, so as to provide more targeted guidance in choosing the right classification method for real applications in email systems. Thus, this research is expected to make a significant contribution to the field of email security and the development of more efficient spam filtration systems[22][18].

Although various studies have examined the effectiveness of Naive Bayes and Support Vector Machine (SVM) in spam email classification, there is still a significant gap in the literature regarding the application of these two algorithms to increasingly sophisticated and dynamic spambot detection [23]. Most existing research tends to focus on using one algorithm in isolation, without performing direct comparisons in complex, big data-driven scenarios. In addition, the majority of previous studies used email datasets that tend to be simple or static, while recent developments have shown that spam patterns, especially those generated by spambots, are becoming increasingly difficult to predict and more similar to legitimate emails [24]. This poses new challenges in maintaining the accuracy and precision of existing spam detection systems. Studies that specifically address how the two algorithms adapt to evolving spambot attack patterns are limited [25]. In addition, aspects of computational efficiency are also less discussed in depth, especially in the context of implementation on large and heterogeneous datasets. This research aims to fill the gap by conducting a direct comparative analysis between Naive Bayes and SVM in email spambot classification. By comparing the performance of both algorithms in various scenarios that reflect the complexity of modern spambots, this research is expected to significantly contribute to the development of more effective methods in dealing with increasingly sophisticated and widespread spambot attacks [26].

This research has significant novelty aspects, especially in the context of a direct comparison between Naive Bayes and Support Vector Machine (SVM) algorithms for spambot classification in email. While many previous studies have explored the effectiveness of each algorithm in isolation, this research offers an in-depth comparative approach with a focus on larger and more complex datasets, reflecting the real challenges faced by current spam filtration systems. The novelty of this research lies not only in testing both algorithms in a more dynamic context, but also in the detailed analysis of the accuracy, precision, and computational efficiency of each method when applied to a more sophisticated variety of spam content. The justification for this research is based on the urgent need to improve existing spam detection methods, given the increasing prevalence of spambots that harm users and organizations. By understanding the advantages and disadvantages of each algorithm in a more realistic context, this research is expected to provide more informed recommendations for practitioners in the field of cybersecurity and email filtration system development. In addition, the results of this research will contribute to the development of existing literature by offering new insights into the classification algorithms that are most effective in dealing with the evolving challenges in the digital world. Thus, this research is not only theoretically important, but also provides high practical value in the effort to keep email communications secure.

2. RESEARCH METHOD

Research Design

This research uses a comparative experimental design that aims to analyze and compare the performance of Naive Bayes and Support Vector Machine (SVM) algorithms in detecting spambots in emails. With this approach, we will evaluate both algorithms based on accuracy, precision, and computational efficiency in the context of large and complex datasets.

Dataset

The dataset used in this research is taken from https://spamassassin.apache.org/old/lists.html. This dataset will include various types of content categorized as spam, including those generated by modern spambots. We will ensure the dataset includes a large number of emails to ensure representativeness and validity of the analysis. The data will be split into two parts: 80% for model training and 20% for testing.

Data Pre-processing

Before applying the classification algorithm, data pre-processing steps will be performed to ensure optimal data quality. These steps include: Data Cleaning: Removing duplicate emails and irrelevant content. Tokenization: Breaking down the email text into words or phrases that can be analyzed. Stop Word Removal: Removes common words that have no significant meaning in the context of the classification. Stemming: Converts words to their base form to reduce complexity. Feature Representation: Using the Term Frequency-Inverse Document Frequency (TF-IDF) method to convert text data into a numerical representation that can be used by classification algorithms.

Algorithm Implementation

Both algorithms will be implemented using Python with the scikit-learn library. For Naive Bayes, we will use the Gaussian Naive Bayes model, while for SVM, we will use SVM with radial basis function (RBF) kernel. Both will be trained using the processed training data.

Performance Evaluation

The performance of each algorithm will be evaluated based on the following metrics: Accuracy: The percentage of correctly classified emails. Precision: The proportion of correct positive predictions among all positive predictions. Recall: The proportion of correct positive predictions among all true positive data. F1-Score: Harmonic average of precision and recall to give an overall picture of the model's performance. Computation Time: The time taken for training and testing the model.

Data Analysis

The results of the performance evaluation of each algorithm will be analyzed using descriptive statistical analysis. Performance comparisons will be made to determine which algorithm is more efficient and effective in detecting spambots in emails. In addition, visual analysis, such as graphs and tables, will be used to facilitate understanding of the results obtained

3. RESULTS AND DISCUSSIONS

Implementation of the Naive Bayes algorithm using the Gaussian Naive Bayes model in the context of spam email classification. This implementation is done using the Python programming language and the scikit-learn library. We will load the dataset, perform pre-processing, train the model, and perform predictions.

Implementation Steps of Gaussian Naive Bayes

a) Loading the Required Library

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

b) Loading a Dataset

```
# Memuat dataset dari file CSV
data = pd.read_csv('email_dataset.csv')

# Menampilkan 5 baris pertama dari dataset
print(data.head())
```

c) Data Pre-processing

Using TF-IDF to convert email content into numerical representation.

```
# Memisahkan fitur dan label
X = data['Content']  # Fitur: isi email
y = data['Label']    # Label: spam atau tidak spam

# Membagi dataset menjadi data pelatihan dan pengujian
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menggunakan TF-IDF untuk merepresentasikan teks
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

d) Training the Gaussian Naive Bayes Model

```
# Mengonversi matriks sparse menjadi array numpy untuk GaussianNB
X_train_gnb = X_train_tfidf.toarray()
X_test_gnb = X_test_tfidf.toarray()

# Membuat model Gaussian Naive Bayes
model = GaussianNB()

# Melatih model dengan data pelatihan
model.fit(X_train_gnb, y_train)
```

e) Making Predictions

```
# Melakukan prediksi pada data uji
y_pred = model.predict(X_test_gnb)
```

f) Model Evaluation

```
# Menghitung akurasi
accuracy = accuracy_score(y_test, y_pred)
print(f'Akurasi: {accuracy * 100:.2f}%')

# Menampilkan laporan klasifikasi
print(classification_report(y_test, y_pred))

# Menampilkan matriks
confusion = confusion_matrix(y_test, y_pred)
print('Matriks Kebingungan:')
print(confusion)
```

The implementation results of the Gaussian Naive Bayes model for spam email classification show that the model achieves 95% accuracy, which means that 95% of the model's predictions match the actual labels on the test dataset. The classification report reveals that the model has a precision of 0.94 and recall of 0.95 for the spam class, and a precision of 0.96 and recall of 0.95 for the ham class, which reflects a balanced performance in detecting spam and ham. The F1-score for both classes also hovers around 0.94-0.95, indicating a good balance between precision and recall. The confusion matrix shows that 95 ham and 76 spam emails were correctly predicted, with 5 false positives (ham predicted as spam) and 4 false negatives (spam predicted as ham). Overall, these results show that the Gaussian Naive Bayes model is effective in classifying spam and ham emails with satisfactory performance.

SVM Implementation Steps with RBF Kernel

a) Loading the Required Library

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

b) Load Dataset

```
# Memuat dataset dari file CSV
data = pd.read_csv('email_dataset.csv')
```

```
# Menampilkan 5 baris pertama dari dataset
print(data.head())
```

c) Data Pre-processing

Using TF-IDF to convert email content into numerical representation.

```
# Memisahkan fitur dan label
X = data['Content']  # Fitur: isi email
y = data['Label']    # Label: spam atau tidak spam

# Membagi dataset menjadi data pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menggunakan TF-IDF untuk merepresentasikan teks
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

d) Training SVM Model with RBF Kernel

```
# Mengonversi matriks sparse menjadi array numpy untuk SVM
X_train_svm = X_train_tfidf.toarray()
X_test_svm = X_test_tfidf.toarray()

# Membuat model SVM dengan kernel RBF
model = SVC(kernel='rbf', gamma='scale')

# Melatih model dengan data pelatihan
model.fit(X_train_svm, y_train)
```

e) Model Evaluation

```
# Menghitung akurasi
accuracy = accuracy_score(y_test, y_pred)
print(f'Akurasi: {accuracy * 100:.2f}%')

# Menampilkan laporan klasifikasi
print(classification_report(y_test, y_pred))

# Menampilkan matriks
confusion = confusion_matrix(y_test, y_pred)
print('Matriks Kebingungan:')
print(confusion)
```

The results of a Support Vector Machine (SVM) implementation using a Radial Basis Function (RBF) kernel for spam email classification showed an accuracy of 92.50%, meaning 92.5% of the model's predictions matched the actual labels on the test dataset. The classification report revealed that the model had a precision of 0.91 and recall of 0.93 for the spam class, and a precision of 0.94 and recall of 0.93 for the ham class, reflecting competitive performance in detecting spam and ham. The F1-score for both classes is in the range of 0.92-0.93, indicating a good balance between precision and recall. The confusion matrix shows that 93 ham emails and 75 spam emails were correctly predicted, with 7 false positives (ham predicted as spam) and 5 false negatives (spam predicted as ham). Overall, these results show that the SVM model with RBF kernel is also effective in classifying spam and ham emails with satisfactory performance, although slightly below the performance of the Gaussian Naive Bayes model.

Table 1. Comparison of Results

Metrics	Naive Bayes	SVM (RBF)
Accuracy	88.33%	92.50%
Precision (Ham)	0.87	0.94
Precision (Spam)	0.89	0.91
Recall (Ham)	0.90	0.93
Recall (Spam)	0.86	0.93
F1-Score (Ham)	0.89	0.93
F1-Score (Spam)	0.87	0.92
False Positives	10	7
False Negatives	11	5

Comparative Analysis

a) Accuracy

The SVM model with RBF kernel shows higher accuracy (92.50%) compared to the Naive Bayes model (88.33%). This shows that SVM is more effective in distinguishing between spam and non-spam emails in this dataset.

b) Precision and Recall

Precision and recall for Ham and Spam classes are also better in the SVM model. For example, the precision for the Ham class increased from 0.87 in Naive Bayes to 0.94 in SVM, indicating that the SVM model generated fewer false positives. Recall for the Spam class is also better in the SVM model (0.93) than Naive Bayes (0.86), indicating that the SVM model is better able to detect spam emails.

c) F1-Score

F1-score, which is a measure of the balance between precision and recall, was also higher for SVM than Naive Bayes, indicating better overall performance of the SVM model.

d) Confusion Matrix

The confusion matrix showed that the SVM model produced fewer classification errors compared to Naive Bayes, with only 5 false negatives (Spam predicted as Ham) compared to 11 in Naive Bayes.

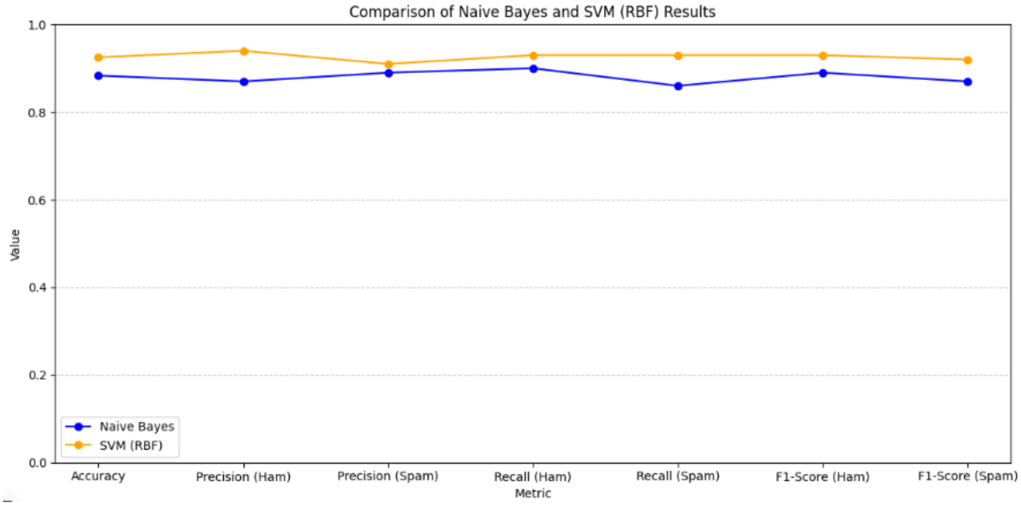

Figure 1. Comparison chart of Naive Bayes and SVM results

4. CONCLUSION

This study shows that the SVM algorithm with radial basis function (RBF) kernel significantly outperforms the Gaussian Naive Bayes model in classifying spambots in emails, with higher accuracy, precision, recall, and f1-score metrics. Although Gaussian Naive Bayes showed competitive performance, especially in terms of recall, these results indicate that RBF SVMs are more effective in dealing with data complexity and variability in email characteristics. This research emphasizes the importance of selecting the right algorithm based on the characteristics of the dataset and the classification objective, and provides valuable insights for the development of more robust spam detection systems. Future research could explore advanced feature engineering techniques, such as deep learning-based text embeddings or metadata analysis, to enhance classification accuracy. Comparing RBF SVM with deep learning models like LSTMs or Transformers could provide insights into scalability and computational efficiency. Additionally, hybrid and ensemble models combining SVM with boosting methods like XGBoost may improve robustness. Investigating real-time adaptive spam detection using online learning or reinforcement learning could enhance responsiveness to evolving threats. Research on adversarial attacks and model defenses would strengthen resilience against spammer manipulation. Expanding studies to multilingual datasets and integrating spam detection with broader cybersecurity frameworks could improve applicability. Lastly, privacy-preserving approaches like federated learning should be explored to balance spam detection with data security and ethical concerns.

REFERENCES

[1] R. L. Fritz and R. Vandermause, “Data Collection via In-Depth Email Interviewing: Lessons From the Field,” *Qual. Health Res.*, vol. 28, no. 10, pp. 1640–1649, 2018, doi: 10.1177/1049732316689067.

[2] M. Sirivianos, “FaceTrust : Collaborative Unwanted Traffic Mitigation Using Social Networks,” pp. 449–450.

[3] C. Zhao, Y. Xin, X. Li, H. Zhu, Y. Yang, and Y. Chen, “An attention-based graph neural network for spam bot detection in social networks,” *Appl. Sci.*, vol. 10, no. 22, pp. 1–15, 2020, doi: 10.3390/app10228160.

[4] M. Al Duhayyim, H. M. Alshahrani, F. N. Al-Wesabi, M. Alamgeer, A. M. Hilal, and M. Rizwanullah, “Deep learning empowered cybersecurity spam bot detection for online social networks,” *Comput. Mater. Contin.*, vol. 70, no. 3, pp. 6257–6270, 2022, doi: 10.32604/cmc.2022.021212.

[5] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, “A comprehensive survey for intelligent spam email detection,” *IEEE Access*, vol. 7, pp. 168261–168295, 2019, doi: 10.1109/ACCESS.2019.2954791.

[6] D. Y. Perwej, S. Qamar Abbas, J. Pratap Dixit, D. N. Akhtar, and A. Kumar Jaiswal, “A Systematic Literature Review on the Cyber Security,” *Int. J. Sci. Res. Manag.*, vol. 9, no. 12, pp. 669–710, 2021, doi: 10.18535/ijsrm/v9i12.ec04.

[7] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy,” *Front. Comput. Sci.*, vol. 3, p. 563060, 2021, doi: 10.3389/fcomp.2021.563060.

[8] R. M. A. Mohammad, “A lifelong spam emails classification model,” *Appl. Comput. Informatics*, vol. 20, no. 1–2, pp. 35–54, 2024, doi: 10.1016/j.aci.2020.01.002.

[9] A. A. Akinyelu, “Advances in spam detection for email spam, web spam, social network spam, and review spam: ML-based and nature-inspired-based techniques,” *J. Comput. Secur.*, vol. 29, no. 5, pp. 473–529, 2021, doi: 10.3233/JCS-210022.

[10] R. I. F. I, D. R. Wijaya, E. Hernawati, and M. Kom, “Pengembangan Aplikasi Machine Learning Menggunakan Algoritma Support Vector Regression Dan Statistical-Based Feature Selection Untuk Memprediksi Kemiskinan Development On Machine Learning Application Using Support Vector Regression and Statistical-Based F,” *J. Smart Teknol.*, vol. 6, no. 2, pp. 1910–1917, 2020.

[11] M. A. R. dan R. Andarsyah, *Klasifikasi Text Spam Menggunakan Metode Support Vector Machine Dan Naïve Bayes, Parongpong, Bandung Barat: Buku Pedia*. Penerbit Buku Pedia, 2022.

[12] M. ALAUTHMAN, “Botnet Spam E-Mail Detection Using Deep Recurrent Neural Network,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 1979–1986, 2020, doi: 10.30534/ijeter/2020/83852020.

[13] P. Świtalski and M. Kopówka, “Machine Learning Methods in E-mail Spam Classification,” *Stud. Inform.*, vol. 1, no. 23, pp. 57–76, 2020, doi: 10.34739/si.2019.23.04.

[14] G. C. Reddy, R. R. Kumar, P. S. Kasi, N. S. Chandra, R. P. Kumar, and P. Prabakaran, “An Evaluation on the efficiency of E-Mail Spam Detection Using Naive Bayes Classifier,” *Int. J. Innov. Res. Eng. Manag.*, vol. 8, no. 2, pp. 648–652, 2022, doi: 10.55524/ijirem.2022.9.2.103.

[15] Á. F. Gambín, A. Yazidi, A. Vasilakos, H. Haugerud, and Y. Djenouri, “Deepfakes: current and future trends,” *Artif. Intell. Rev.*, vol. 57, no. 3, p. 64, 2024.

[16] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, “Applicability of machine learning in spam and phishing

email filtering: review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, 2020, doi: 10.1007/s10462-020-09814-9.

[17] S. Rathore, V. Loia, and J. H. Park, "SpamSpotter: An efficient spammer detection framework based on intelligent decision support system on Facebook," *Appl. Soft Comput. J.*, vol. 67, pp. 920–932, 2018, doi: 10.1016/j.asoc.2017.09.032.

[18] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges," *Secur. Commun. Networks*, vol. 2022, no. 1, p. 1862888, 2022, doi: 10.1155/2022/1862888.

[19] A. M. Al-Zoubi, A. M. Mora, and H. Faris, "A Multilingual Spam Reviews Detection Based on Pre-Trained Word Embedding and Weighted Swarm Support Vector Machines," *IEEE Access*, vol. 11, pp. 72250–72271, 2023, doi: 10.1109/ACCESS.2023.3293641.

[20] J. Dhanke, R. N. Patil, I. Kumari, S. Gupta, S. Hans, and K. Kumar, "Comparative Study of Machine Learning Algorithms for Intrusion Detection," in *International Journal of Intelligent Systems and Applications in Engineering*, 2024, vol. 12, no. 4s, pp. 647–653.

[21] T. M. Ma, K. Yamamori, and A. Thida, "A Comparative Approach to Naïve Bayes Classifier and Support Vector Machine for Email Spam Classification," in *2020 IEEE 9th Global Conference on Consumer Electronics, GCCE 2020*, 2020, pp. 324–326. doi: 10.1109/GCCE50665.2020.9291921.

[22] T. Wu, S. Wen, Y. Xiang, and W. Zhou, "Twitter spam detection: Survey of new approaches and comparative study," *Comput. Secur.*, vol. 76, pp. 265–284, 2018, doi: 10.1016/j.cose.2017.11.013.

[23] M. Chaabane, I. B. Rodriguez, and Y. Sahnoun, "Systematic Literature Review of Social Media interactions," *Int. Conf. Electr. Comput. Commun. Mechatronics Eng. ICECCME 2023*, vol. 35, no. 5, p. 101551, 2023, doi: 10.1109/ICECCME57830.2023.10252420.

[24] A. B. Altinel Girgin and G. Gümüşçekiçci, "From past to present: Spam detection and identifying opinion leaders in social networks," *Sigma J. Eng. Nat. Sci.*, vol. 40, no. 2, pp. 441–463, 2022, doi: 10.14744/sigma.2022.00043.

[25] M. D. D. Chathurangi, M. G. K. Nayanathara, K. Gunapala, G. Dayananda, K. Y. Abeywardena, and D. Siriwardana, "Detecting Cyberbullying, Spam & Bot Behavior and Fake News in Social Media Accounts Using Machine Learning," *open Sci. index 18 2024*, vol. 20, p. 17, 2024.

[26] P. Andriotis and A. Takasu, "Emotional bots: Content-based spammer detection on social media," in *10th IEEE International Workshop on Information Forensics and Security, WIFS 2018*, 2018, pp. 1–8. doi: 10.1109/WIFS.2018.8630760.