

Email Spam Detection Using Machine Learning

Dr. T. C. Ezhil Selvan¹, M.Diviyasri², R.Dharshini³

¹ Assistant Professor, Dept of Information Technology

^{2,3}Dept of Information Technology

^{1,2,3} Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu

Abstract- In view of the current COVID-19 situation, the globe has shifted to online communication, leading to a notable rise in data exchange using platforms like email and social media. Ensuring the safety of these digital exchanges is crucial in the realm of computer security. With the continual growth of the Internet, email has emerged as a trustworthy and efficient way to share data. Nowadays, most interactions, whether personal or business-related, are carried out via email as the primary form of correspondence. Email provides the benefit of instant communication, saving time and money. However, despite its numerous advantages, email communication has been susceptible to spam assaults. Spam emails are frequently employed to flood mailboxes with large volumes of messages, causing inconvenience and clutter for recipients. To tackle this issue, this research aims to assess the effectiveness of Support Vector Machine (SVM) techniques in categorizing email data, specifically focusing on detecting spam emails. The SVM algorithms will be compared to determine which one achieves optimal accuracy in organizing emails. The Support Vector Machine tactic is recognized for its efficiency, accuracy, and simplicity in deploying the proposed algorithm. It aids in verifying the semantic content of emails and can prevent the unnecessary flow of unimportant messages. The research will be conducted using the Python programming language, and the outcomes will shed light on the most efficient technique for email sorting and spam identification.

Keywords- Online communication, Data exchange, Computer security, Spam emails, Support Vector Machine (SVM), Email categorization, Spam detection, Email sorting, Accuracy.

I. INTRODUCTION

Email is a highly effective method of online communication due to its cost-saving benefits and time-efficiency, making it a preferred medium for both personal and professional interactions. Business correspondence through emails streamlines data transfer, enabling the transmission of various files globally. However, emails are occasionally vulnerable to multiple attacks, either through active or interactive means. Users may receive messages from unfamiliar sources or encounter irrelevant content, known as spam mails, which inundate email accounts with unwanted or

excessive data. Spam mails are notorious for inundating unique or random email addresses with unsolicited and repetitive messages, some of which may contain malicious software or executable files that can compromise a user's system security. The majority of email and spam lists are compiled by scouring Usenet advertisements and illicitly obtaining Internet email listings. An email is deemed spam if it meets three criteria: anonymity, mass mailing to a large audience, and unsolicited nature.

II. MACHINE LEARNING CLASSIFICATION ALGORITHMS

Naive Bayes: Naive Bayes is an algorithm for classification that is appropriate for both binary and multiclass classification. Naive Bayes exhibits better performance with categorical input variables compared to numerical variables, making it valuable for making predictions based on past outcomes and projected data.

$$p(A|B)=p(b|A)p(A)/p(B)$$

P(A) is for Prior Probability: The chance of a hypothesis before observing the evidence.

P(B) is for Marginal Probability: The likelihood of Evidence.

Support Vector Machine: SVMs find application in intrusion detection, face recognition, email categorization, gene grouping, and webcontent categorization. They can handle both linear and non-linear data for classification and regression tasks.

Decision tree: Decision trees are extremely beneficial for data analysis and machine learning as they simplify complex data into more manageable sections. They are typically employed in predictive analysis, data categorization, and regression tasks. Entropy using the frequency table of one attribute and Entropy using the frequency table of two attributes.

KNN: The KNN algorithm can rival highly precise models due to its accurate predictions. KNN is utilized for tasks requiring high precision without the need for a human-readable model.

The prediction quality depends on the distance measurement.
Formula:

$$\text{dist}((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$$

Random forest classifiers: Random forest classifiers can tackle regression or classification issues. This algorithm consists of various decision trees, each tree comprising data samples drawn from the training set with replacement, known as bootstrap samples.

III. PROBLEM STATEMENT

Spam has experienced a significant surge in recent times, constituting around 70% of all email traffic. Reflecting the trend seen with internet expansions, the issue of spam email is on the rise too. A study discovered that close to 10 days per year were compromised due to spam handling. The costly nature of spam poses a substantial financial burden on bandwidth providers for the foreseeable future. Continuous efforts are needed to combat spam's persistence in the realm of email communication. Hence, it remains imperative to differentiate between unwanted emails and leverage recommended techniques for distinguishing spam emails from legitimate ones. Such approaches are vital for accurate email classification and algorithmic efficiency. The machine learning process demonstrates notable rapidity. Multiple algorithms are employed for unsolicited email categorization, with Support Vector Machine standing out for its efficacy. This algorithm plays a decisive role in categorizing neural networks for improved classification. Implementing the Support Vector Machine enables precise results in email categorization. To enhance the categorization function's outcomes, an attribute selection algorithm must be applied to the dataset. The primary selection approach utilized here is the Best-First algorithm, which refines the entity and category selection processes. Through this study, it was evident that choosing a Best-First algorithm-based feature led to superior category identification over other classifiers. Consequently, the dataset optimization enhances the performance of Support Vector Machine, culminating in improved classification precision.

IV. CURRENT SYSTEM

Existing methods for filtering email spam involve List-Based Filter techniques, such as Blacklist, Real-time Blackhole List, Whitelist, and Grey list. Blacklists are commonly used to combat unwanted emails by filtering messages from a predetermined list of senders created by the organization's system administrator. These blacklists consist of email addresses or Internet Protocol (IP) addresses that have previously been associated with sending spam. Upon receiving a new message, the spam filter checks if its IP or

email address matches any entry on the blacklist; if there is a match, the message is marked as spam and discarded. While blacklists prevent known spammers from reaching users' inboxes, they can also misclassify legitimate senders as spammers. This can occur when a spammer sends unsolicited emails from an IP address shared by authentic email users.

V. EXISTING SYSTEM

A Naive Bayes classifier is a straightforward probabilistic classifier that makes strong assumptions of independence. In simple terms, it operates on the idea that the presence or absence of a specific property within a class is unrelated to the presence or absence of any other attribute. This notion considers the class variable as dependent on the Class Probability Model, which is trained in a supervised learning setting. An advantage of naive Bayes classification is its minimal requirement for training data to estimate the necessary parameters for classification. This classification method underlines the belief that the data correlates with a particular class, followed by the calculation of the likelihood of that assertion. Essentially, Bayesian classifiers are statistical tools capable of predicting probabilities of class inclusion, such as the likelihood of a given sample belonging to a specific class.

VI. PROPOSED SYSTEM

The proposed system aims to boost the email classification process through the utilization of advanced machine learning tactics, specifically Support Vector Machines (SVM) and Naïve Bayes (NB). These algorithms have the ability to assess and categorize emails by studying a broad dataset of categorized instances, heightening the precision in distinguishing between legitimate and unwanted messages. By employing SVM and NB, the system can adjust to evolving spamming trends, thus increasing its resistance to novel spamming methods and reducing false positive outcomes. Additionally, the Python programming language will be employed for the development and execution of the proposed system, presenting a more efficient and expandable approach for email arrangement.

This strategy will guarantee that users obtain a more dependable and secure email experience, empowering them to effectively filter out spam while accurately recognizing and transmitting essential messages to their inbox.

VII. METHODOLOGY

Data Collection: Collect a diverse array of email data, including both legitimate (ham) and spam emails. The dataset

should cover various sources, formats, and types of content to ensure thorough training and testing.

Data Pre-processing: The email data should be preprocessed by eliminating HTML tags, special characters, and irrelevant details. Tokenize the text and conduct stemming or lemmatization to standardize the language.

Feature Extraction: identify pertinent features from the email content, like term frequency-inverse document frequency (TF-IDF) values, word embeddings, and metadata such as sender information. These features will serve as inputs for the machine learning models.

Data Split: Segment the dataset into training and testing subsets with a proper distribution of ham and spam emails in each set. Consider using cross-validation for a more rigorous model evaluation.

Model Selection: Deploy and train Support Vector Machines (SVM) and Naïve Bayes (NB) classifiers with the training data. Adjust hyperparameters and evaluate the models' performance based on metrics like accuracy, precision, recall, and F1-score.

Model Evaluation: Assess the SVM and NB models on the testing dataset to gauge their efficiency in classifying email messages. Compare their accuracy and resilience against spam threats.

Model Integration: Incorporate the selected machine learning model (SVM or NB) into the email sorting system to automate the process of categorizing incoming emails as ham or spam.

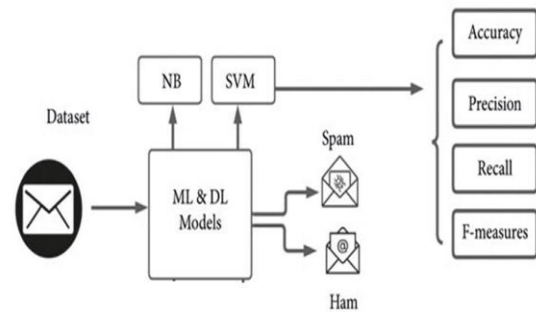
Real-Time Implementation: Integrate the system into real-time email platforms to ensure efficient email sorting and spam detection.

Performance Optimization: Regularly monitor the system's performance and adjust the model as necessary to adapt to evolving spamming tactics and changing email content.

User Feedback: Solicit user feedback to enhance the system's accuracy and user satisfaction. This feedback can help refine the model and improve its spam detection capabilities.

Deployment: Launch the upgraded email classification system, enhancing the existing rule-based filters, and ensuring regular updates to sustain its effectiveness in distinguishing ham and spam emails.

VIII. BLOCK DIAGRAM



Block Diagram for Spam Detection

IX. CLASSIFICATION

9.1 Types of support vector machines

Support vector machines come in various forms with distinct functionalities tailored to different problem scenarios. Presented below are two categories of SVMs and their respective significance:

1. Linear SVM.

Linear SVMs use a linear kernel to establish a direct decision boundary, segregating classes effectively. They excel when data can be separated linearly or when a linear approximation suffices. This category of SVMs is computationally efficient and offers a high level of interpretability due to the decision boundary being a hyperplane within the feature space.

2. Nonlinear SVM.

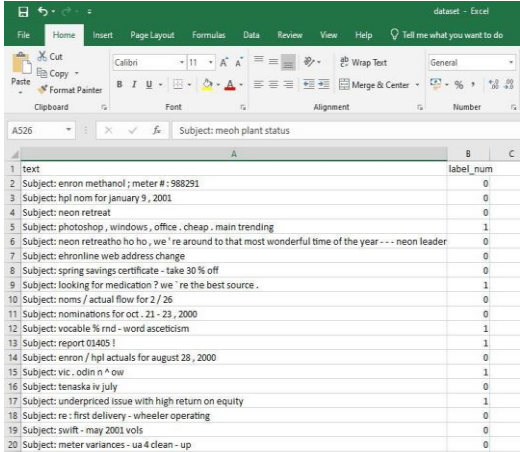
Nonlinear SVMs cater to situations where data cannot be divided by a straight line in the feature space. They accomplish this by leveraging kernel functions that map the data implicitly into a higher-dimensional feature space. Within this space, a linear decision boundary can be located. Kernel functions commonly utilized in Nonlinear SVMs include the polynomial, Gaussian (RBF), and sigmoid kernels. These SVMs excel in capturing intricate patterns, leading to higher accuracy in classification compared to their linear counterparts.

9.2 Proposed Algorithm

- Step 1: Select the email
- Step 2: Utilize tokenization and word count algorithms.
- Step 3: Employ the Support Vector Machine Classifier.
- Step 4: Testing the dataset

X. WORKING OF THE PROPOSED SYSTEM

The primary goal of this venture is to improve the security of email communication through the creation of a system that classifies emails and detects spam. The system's operation involves a clearly defined approach starting with the compilation of a varied dataset of email messages, including both legitimate (ham) and spam emails. These emails are then preprocessed to eliminate irrelevant data and standardize the text. Important characteristics are subsequently derived from the email contents, such as term frequency-inverse document frequency (TF-IDF) values and sender metadata. The dataset is divided into training and testing subsets to evaluate the model.



```

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve, auc
import matplotlib.pyplot as plt
import seaborn as sns
import joblib

# Load your unbalanced dataset
data = pd.read_csv('dataset.csv')

# Separate features and labels
X = data['text']
y = data['label_num']

# TF-IDF Vectorization of text data
vectorizer = TfidfVectorizer(max_features=5000) # Adjust max_features as needed
X_transformed = vectorizer.fit_transform(X)

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

# Initialize and train an SVM classifier
svm_classifier = SVC(kernel='linear', C=1)
svm_classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = svm_classifier.predict(X_test)
    
```

Fig 10.1.1 Splitting of dataset

Support Vector Machine (SVM) classifiers undergo implementation and fine-tuning, followed by a comparison of their performance using different metrics. The selected classifier is then integrated into email platforms in real-time to automate the sorting of emails and detect spam. This integration guarantees users a dependable, effective, and secure experience in email communication. Moreover, the

system offers the adaptability to counter new spam techniques and changing email content, making it a crucial tool for the digital era.

```

class_report = classification_report(y_test, y_pred)
print("Classification Report:")
print(class_report)

# Calculate ROC curve and AUC using the trained SVM classifier
y_prob = svm_classifier.decision_function(X_test) # Get decision function values for ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc='lower right')
plt.show()
    
```

Fig 10.1.2 True and False Positive rate

Receiver working illustrates the true positive and false positive rate. Electronic mail, with its abilities for real-time communication and cost-effectiveness, has emerged as a key method of sharing information, both in personal and business settings. Nevertheless, the effectiveness of email communication has been impaired by spam attacks, inundating mailboxes with unsolicited messages and causing significant inconvenience to users. In this project, an effort is made to tackle this problem by creating an email classification and spam detection system that utilizes machine learning techniques

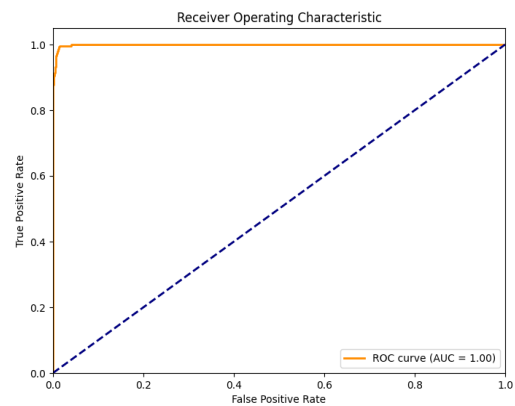


Fig 10.1.3 Receiver Operating Characteristic

The main goals of the project involve creating a reliable email categorization and spam identification system, evaluating the efficacy of Support Vector Machine, and ensuring users access a safe and efficient email interaction that adjusts to changing email patterns and emerging spam techniques. This thorough approach and the classifiers selected collectively contribute to enhancing email security in the digital age.

The email sorting and spam spotting system provides various key benefits, such as its capability to effectively and precisely differentiate between legitimate and spam messages. Moreover, the system is expandable, ensuring its adaptability to an increasing count of email variables and data elements.

Its capability to provide instantaneous and accurate predictions, coupled with its strength against irrelevant attributes, heightens its overall performance. This endeavor opens up substantial possibilities for future improvements and expansions. These options include investigating sophisticated attribute manipulation techniques like word embeddings and deep learning structures, creating combined classification systems amalgamating the advantages of both Support Vector Machine classifiers, and building real-time monitoring to adjust to changing spam strategies. Additionally, integrating user feedback, supporting multiple languages, enhancing security elements, incorporating cloud functionality, and implementing visualization and reporting tools can all heighten the system.

By using the training data set in Support Vector Machine algorithm, the accuracy level of the system is predicted as 0.98%. with two different value 0 and 1 as spam and ham. the classification report shows the precisions, recall, f1 score, support of the model. Thereare total 1035 mails.For the enhancement in user interaction,the program incorporates a graphical user interface (GUI) developed with Tkinter. This GUI allows user to load a test dataset and initiate the classification process through a user-friendly interface. The interface features button to open a test files, start the classification, and loading label indicating the process of classification. This user-friendly design ensures accessibility andeasyof use for individuals for classification the data uploaded screen is shown in the figure

XI. RESULTS

Output of Proposed System

The outcomes from our tests show that the Support Vector Machine has the ability to effectively categorize emails, with its performance being influenced by the data's characteristics and specific usage scenarios. The Support Vector Machine shines in tasks related to sorting text and documents, showcasing its robustness in segregating both linear and non-linear data.

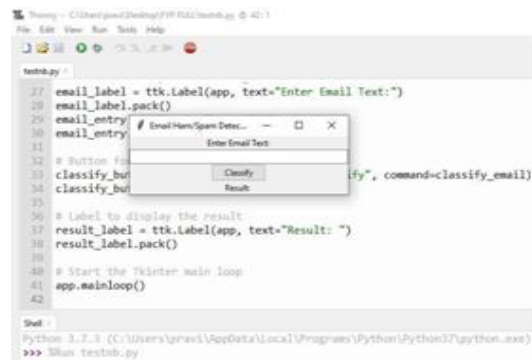


Fig 11.1.3 User Input for Training

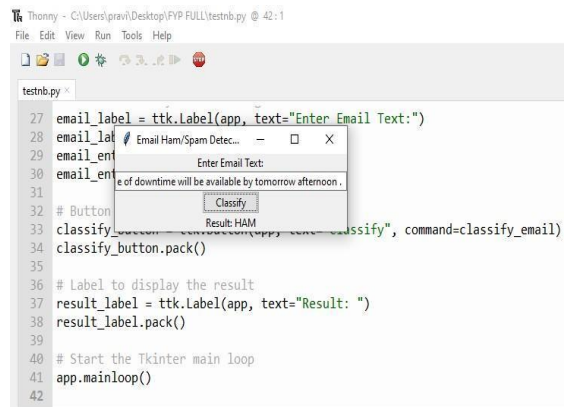


Fig 11.1.4 Detecting Ham emails

```

main2.py
1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3 from sklearn.svm import SVC
4 from sklearn.model_selection import train_test_split
5 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8 import joblib
9
10 # Load your unbalanced dataset
11 data = pd.read_csv('dataset.csv')
12
13 # Separate features and labels
14 X = data['text']
15 y = data['label_num']
16
17 # TF-IDF Vectorization of text data
18 vectorizer = TfidfVectorizer(max_features=5000) # Adjust max_features as needed
19 X_transformed = vectorizer.fit_transform(X)
20
21 # Train SVM model
22 X_train, X_test, y_train, y_test = train_test_split(X_transformed, y, test_size=0.2)
23
24 # Train the SVM model
25 svm = SVC()
26 svm.fit(X_train, y_train)
27
28 # Save the trained model
29 joblib.dump(svm, 'svm_model.pkl')
30
31 # Test the model
32 accuracy = accuracy_score(y_test, svm.predict(X_test))
33 print(f'Accuracy: {accuracy}')
34
35 # Confusion Matrix and Classification Report
36 cm = confusion_matrix(y_test, svm.predict(X_test))
37 report = classification_report(y_test, svm.predict(X_test))
38 print('Confusion Matrix:')
39 print(cm)
40 print('Classification Report:')
41 print(report)
42
Shell -
Accuracy: 0.9874396135265701

```

Fig 6.1.1 Accuracy of Training data

```

Accuracy: 0.9874396135265701
Classification Report:

```

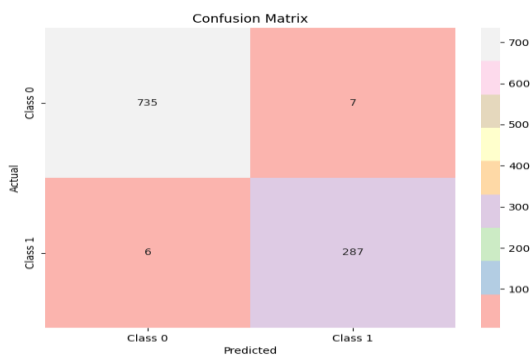
	precision	recall	f1-score	support
0	0.99	0.99	0.99	742
1	0.98	0.98	0.98	293
accuracy			0.99	1035
macro avg	0.98	0.99	0.98	1035
weighted avg	0.99	0.99	0.99	1035

```

Confusion Matrix:
[[735  7]
 [ 6 287]]

```

Fig 11.1.2 Output of training data



11.2 Confusion Matrix

REFERENCES

- [1] A. Karim, S. Azam, B. Shanmugam, K. Kannurpatti, 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.
- [2] E. Bauer. 15 Outrageous Email Spam Statistics That Still Ring True in 2018, RSS. Accessed: Oct. 10, 2020. [Online]. Available: <https://www.propellercrm.com/blog/email-spam-statistics>
- [3] D. Lynkova. (2021). How Many Emails are Sent Per Day: The Startling Truth [2021], TechJury. Accessed: Sep. 23, 2020. [Online]. Available: <https://techjury.net/blog/how-many-emails-are-sent-per-day>
- [4] K. Sheridan. (2020). FBI: Business Email Compromise Cost Businesses 1.7B in 2019, Dark Reading. Accessed: Mar. 21, 2021. [Online]. Available: <https://www.darkreading.com/fbi-business-email-compromise-cost-businesses-17b-in-2019/d-d-id/1337035>
- [5] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *ArtifIntell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020, doi:10.1007/s10462-020-09814-9.
- [6] J. Johnson. (2021). Spam email: Countries of origin 2020. Statista. accessed: Nov. 27, 2020. [Online]. Available: <https://www.statista.com/statistics/263086/countries-of-origin-of-spam>
- [7] A. Karim, S. Azam, B. Shanmugam, and K. Kannoorpatti, "Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework," *IEEE Access*, vol. 8, pp. 154759–154788, 2020, do 10.1109/access.2020.3017082.
- [8] O. Alonso, "Challenges with label quality for supervised learning," *J. Data Inf. Qual.*, vol. 6, no. 1, pp. 1–3, Mar. 2015.
- [9] S. Manlangit, "Novel machine learning approach for analyzing anonymous credit card fraud patterns," *Int. J. Electron. Commerce Stud.*, vol. 10, no. 2, pp. 175–202, Dec. 2019, doi: 10.7903/ijecs.1732.
- [10] M. Basavaraju and D. R. Prabhakar, "A novel method of spam mail detection using text based clustering approach," *Int. J. Comput. Appl.*, vol. 5, no. 4, pp. 15–25, Aug. 2010, doi: 10.5120/906-1283.
- [11] R. M. Ravindran and D. A. S. Thanamani, "K-means document clustering using vector space model," *Bonfring Int. J. Data Mining*, vol. 5, no. 2, pp. 10–14, Jul. 2015, doi: 10.9756/bijdm.8076.
- [12] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Inf. Sci.*, vol. 277, pp. 421–444, Sep. 2014, doi: 10.1016/j.ins.2014.02.114.
- [13] Y. Cabrera-León, P. G. Báez, and C. P. Suárez-Araujo, "Self-organizing maps in the design of anti-spam filters—A proposal based on thematic categories," in *Proc. 8th Int. Joint Conf. Comput. Intell.*, 2016, pp. 21–32, doi:10.5220/0006041400210032.
- [14] H. Padhiyar and P. Rekh, "An improved expectation maximization based semi-supervised email classification using Naïve Bayes and K-nearest neighbor," *Int. J. Comput. Appl.*, vol. 101, no. 6, pp. 7–11, Sep. 2014, doi:10.5120/17689-8652.
- [15] D. Hao, L. Zhang, J. Sumkin, A. Mohamed, and S. Wu, "Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2701–2710, Sep. 2020, doi: 10.1109/jbhi.2020.2974425.
- [16] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 367–368, Jun. 2010, doi: 10.1145/1811099.1811090.
- [17] G. Dougherty, "Unsupervised learning," in *Pattern Recognition and Classification*. New York, NY, USA: Springer, 2012, pp. 143–155, doi: 10.1007/978-1-4614-5323-9_8.
- [18] S. Russell and P. Norvig, "A modern, agent-oriented approach to introductory artificial intelligence," *ACM SIGART Bull.*, vol. 6, no. 2, pp. 24–26, Apr. 1995, doi: 10.1145/201977.201989