A PROJECT REPORT

on

“Spam Email Classification with Comparative Algorithm Analysis”

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

SAAHEN SRIYAN MISHRA	21051080
ABHISEK SAHOO	22057002
ANJALI PANDA	22057014

UNDER THE GUIDANCE OF
MR. CHANDRA SHEKHAR

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
April 2024

A PROJECT REPORT

on

“Spam Email Classification with Comparative Algorithm Analysis”

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

SAAHEN SRIYAN MISHRA	21051080
ABHISEK SAHOO	22057002
ANJALI PANDA	22057014

UNDER THE GUIDANCE OF
MR. CHANDRA SHEKHAR

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAE, ODISHA -751024
April 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

“Spam Email Classification with Comparative Algorithm Analysis”

submitted by

SAAHEN SRIYAN MISHRA	21051080
ABHISEK SAHOO	22057002
ANJALI PANDA	22057014

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Technology (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 07/ 04/ 2024

(Mr. Chandra Shekhar)
Project Guide

Acknowledgment

We are profoundly grateful to **MR. CHANDRA SHEKHAR** of **KIIT DU** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

SAAHEN SRIYAN MISHRA
ABHISEK SAHOO
ANJALI PANDA

ABSTRACT

Considering various forms of communication, emails represent the optimal medium for both casual and formal dialogues. They are commonly employed for exchanging important information, including text, images, and documents, among individuals using electronic devices. However, the influx of unwanted bulk emails, particularly commercial ones, can impede mailbox storage. Over the years, the proliferation of spam emails has led to the development of numerous spam detection methods. Typically, emails are categorized as either "Ham" or "Spam," aiding in their segregation. This classification process is crucial for bolstering email security. This extensive project primarily focuses on comparing different machine learning approaches for detecting spam emails. Our goal is to identify the most effective algorithmic framework capable of accurately distinguishing between spam and legitimate emails, thereby safeguarding users' digital communication channels against spam infiltration.

The research encompasses a wide range of popular machine learning algorithms, including the simplicity of Naive Bayes, the robustness of Support Vector Machines (SVM), the ensemble techniques of Random Forest, and the anomaly detection capabilities of Isolation Forest. We utilize a carefully curated dataset of labeled email samples, which undergo preprocessing to convert textual content into numerical representations suitable for machine learning models. Through thorough experimentation and evaluation on a separate test dataset, we assess performance metrics such as accuracy, precision, recall, and F1-score to determine the efficacy of each algorithm. Additionally, insights gained from analyzing misclassified emails and examining model behavior aid in selecting the most suitable algorithm for spam email classification. This comprehensive comparative analysis not only enhances email security measures but also contributes to the advancement of spam detection techniques, thus reinforcing users' protection against email-based threats.

Keywords: Spam-Detection, Classification-accuracy, precision, recall, F1-score, Kappa statistic.

Contents

List of Figures

List of Tables

Chapter 1

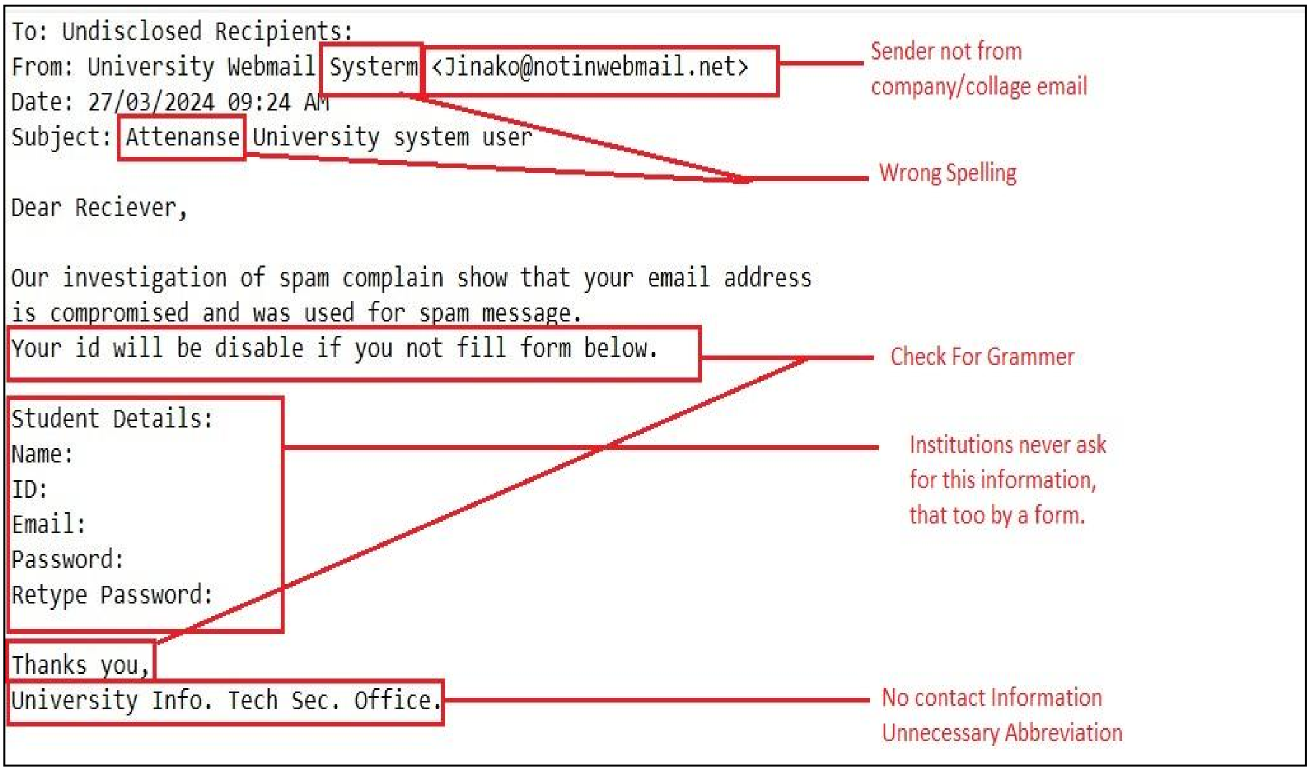
Introduction

In practice, many approaches have been proposed that could identify spam from a mailbox using machine learning methods. Unsolicited bulk emails, especially commercial emails, affect the memory capacity of the inbox. It would be difficult for the user to manually delete all the unwanted or unused mail. Several spam detection methods have been developed to deal with this problem as the spam problem has grown over the years. Generally, all emails are classified as "Ham" or "Spam". Ham messages are targeted or secure legitimate messages in the mailbox, while spam is junk mail, unsolicited bulk messages or business letters in the mailbox. Such filtering or categorization of emails into Ham and Spam helps to separate them from each other, remove spam using automation. There are usually several parameters or components that help detect spam. We may be considered spam if it contains bad grammar, distorted images, distorted symbols or logos, bad links, tempting offers and time-based subscriptions that force users to subscribe now.

Phishing is also considered as one of the dangerous cybercrimes that target individuals and trick them into clicking on links or ordering personal information such as login details for social accounts such as Twitter, Facebook or online banking. in the worst. Phishing messages are also considered spam. Spam also includes spam websites that advertise products that contain URLs that redirect to other websites, 419 Scams - spam that offers users a small initial payment of a large amount, Image Spam - the content of the information displayed in the email is shown in form. . of images. This can also be prevented manually by unsubscribing from emails, using secure email readers/software (eg g-mail, yahoo, outlook, etc.), installing security software and keeping it always up to date. However, it is not very easy because sometimes important or useful data can be deleted and cannot be recovered. Email spam filtering is one of the commonly used processes that help organize all emails according to defined criteria. This process is automated because it automatically arranges all emails according to conditions when they arrive at the inbox server. These spam filtering approaches do not follow any rules and regulations. To further improve it, it can be trained, which helps it learn from previously grouped or classified spam or ham messages. This improvement is called classification, which involves the training and filtering processes of a given email dataset [1].

Data can vary in form, sometimes comprising a mix of images, texts, videos, etc., which cannot be directly used as a classifier. All these classification problems should be taken into account to fully define a classifier. Consumption of memory capacity of servers, with additional costs either to the user, the service provider or the company, even if they are not fully used for the period when the spamming begins and requires the purchase of additional storage space. In addition, its storage capacity grows exponentially, as millions of carriers use the same email client. It is very easy for the user to ignore or accidentally delete messages, which can be important if normal messages end up in spam. The reality of spam plagues businesses at all stages because critical communication at every level of the organization depends on email. Spam filters can reduce the number of unsolicited messages to the lowest possible limit. Email filtering is the collection of emails according to such requirements in order to reorder them. These filters are usually related to the processing of incoming e-mails, scanning, monitoring and deleting e-mails that contain malicious files such as viruses, trojans or ransomware.

Some protocols, such as SMTP, affect email functionality. Spam filtering is located in important places on both consumers and servers. Spam filtering is implemented by many ISPs at each network layer, before the mail server or in the mail if there is a firewall.



To: Undisclosed Recipients:
From: University Webmail Systerm <Jinako@notinwebmail.net>
Date: 27/03/2024 09:24 AM
Subject: Attenanse University system user

Dear Reciever,

Our investigation of spam complain show that your email address
is compromised and was used for spam message.
Your id will be disable if you not fill form below.

Student Details:
Name:
ID:
Email:
Password:
Retype Password:

Thanks you,
University Info. Tech Sec. Office.

Sender not from company/collage email

Wrong Spelling

Check For Grammer

Institutions never ask for this information, that too by a form.

No contact Information
Unnecessary Abbreviation

Figure 1.1: Basis for Identification of Spam Emails

The Email Server is a built-in anti-spam and anti-virus tool that provides strong email protection at the edge of the network. Filters can be enabled as external inputs on computers between certain terminals. These filters can be used in clients. Unsolicited or questionable emails are blocked by filtering, which threatens the security of the network by accessing the operating system. Several popular platforms such as Outlook, Gmail and Yahoo have added various filters to filter out spam so that customers can forward legitimate emails. Contrary to this situation, these filters can also wrongly block legitimate emails. Email companies have created various frameworks for using spam filters. Frames are used to assess the risk level of each email received. Cases include enforcement of spam restrictions, sender protections, black and white lists and means of monitoring recipients. One or more clients can use these methods. If spam is low, more spam will be blocked and sent to recipient mailboxes. A very high threshold can exclude certain large emails unless the user redirects them.

Data mining as an emerging field that involves extracting implicit, previously unknown and potentially useful information from data that is researched and used to create. software that automatically scans databases looking for regularities or patterns. Strong patterns are identified and are likely to be used to generalize and make accurate predictions. Data mining typically uses classification or prediction tasks, which are supervised methods that aim to discover hidden relationships between a target class and independent variables. In supervised learning, classifiers allow observations to be labeled so that unobserved data can be classified based on training data. Spam detection systems are built using classification algorithms that groups emails as spam or Ham (Non - Spam) [2].

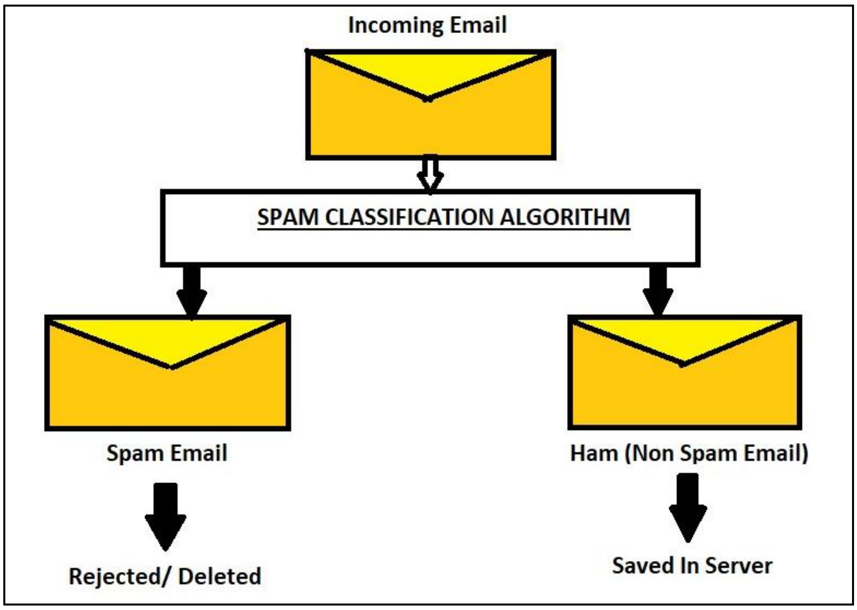


Figure 1.2: Output given by the algorithms for classification.

Chapter 2
Basic Concepts/ Literature Review

The global research community is highly interested in email spam filtering, which has experienced a rapid increase in attention recently. This section discusses similar studies presented in the literature and identifies unresolved issues to highlight discrepancies in the research. Both professional and personal use of emails is acknowledged, with recognition that they are often considered official communication documents. Email analysis and data processing serve various purposes such as subject classification, spam detection, and categorization. It is evident that unsupervised filtering is commonly employed to sift through the extensive body of existing research. While some practices incorporate additional features, they tend to focus on a select few significant aspects of emails, yielding noteworthy results.

2.1 Technical Overview

The proliferation of spam emails poses a significant challenge to global communication systems due to their accessibility to anyone with an internet connection. Various methodologies for mitigating spam involve automated detection of spam-related phrases and the blacklisting of domains associated with such spam. Despite concerted efforts to combat this issue through automated detection and domain blacklisting methods, accurately distinguishing between legitimate emails and spam remains a challenging task [3]. To address this challenge, researchers have turned to machine learning techniques to identify subtle patterns indicative of spam content. This approach goes beyond simple keyword analysis by examining various attributes of emails, such as recipient details, origins, and subject matter. These attributes serve as key components in the classification process. At the core of this methodology lies the Naive Bayes algorithm, a statistical model that plays a central role in classifying emails as either spam or non-spam. Additionally, the Particle Swarm Optimization algorithm is employed to fine-tune parameters, thereby enhancing the accuracy of classification. Experimental validation using established datasets, such as the Ling spam dataset, confirms the effectiveness of this approach. Real-world applications further demonstrate its ability to detect phishing attempts – a particularly harmful form of cyber threat [4].

A broad practical corpus of e-mails pre-marked as spam including phishing, and ham (legitimate) is gathered for methodological assessment. The studies use the techniques to identify phishing emails other published approaches. The system addresses the effect of these effects on the process of incorporating this method into an email provider's system. Eventually, it outlines a plan for updating and adapting filters to detect phishing categories [5].

In a real-world context, the effectiveness of such filters is demonstrated, suggesting the readiness of the technique for deployment. Additionally, novel features are identified for detecting phishing emails, including mathematical models for low-dimensional topic definition, sequential evaluation of emails and external connections, and identification of embedded logos and covert salting indicators. The deliberate inclusion or manipulation of content to evade detection, known as secret salting needed to be addressed as well [6]. The system's resilience to various adversarial effects is assessed, highlighting considerations for integrating this method into an email provider's system. Moreover, research to identify novel techniques for detecting hidden manipulations within emails, such as covert salting is required. By compiling a comprehensive dataset of annotated emails spanning various categories, including spam, phishing, and legitimate communications, empirical analyses should be performed for the proposal of a robust framework.

2.2 Models for the problem

In the ongoing battle against the pervasive problem of spam email, an array of models has been both employed and suggested to bolster detection and mitigation efforts. Among the established models, Bayesian algorithms emerge as a stalwart, leveraging probabilistic principles to effectively classify emails based on learned probabilities of word occurrences. Meanwhile, Support Vector Machines (SVM) offer an alternative approach, utilizing a hyperplane to meticulously segregate emails into distinct categories of spam and non-spam by analyzing their intricate feature vectors [6]. Additionally, the simplicity yet robustness of Naive Bayes cannot be understated, as it assumes independence between features and is frequently deployed for text classification tasks, including spam detection [7]. Furthermore, Rough Set (RS) principles have been strategically integrated into spam filtering systems, establishing rule execution systems that elevate the precision and efficiency of classification .

Innovations in spam detection have led to the proposal of hybrid models like the Bayes-SVM-NB framework, which amalgamates elements of Bayesian algorithms, SVM, and Naive Bayes to yield improvements in accuracy and speed [8]. Moreover, the introduction of the BPNN filter technique showcases a fusion of traditional filtering methods with the capabilities of Backpropagation Neural Network (BPNN) to discern relevant emails from their unsolicited counterparts, thereby augmenting overall filtering efficacy. Recent strides in the field also include the integration of conceptual techniques within spam filters, harnessing advanced methods to comprehensively analyze email content for indications of spam [9]. Furthermore, the continual evolution of machine learning algorithms, such as ensemble techniques like Adaboost, serves to further enhance spam detection accuracy and adaptability to emerging threats.

However, amidst these advancements, challenges persist, necessitating a concerted effort to address evolving spam tactics, optimize scalability and efficiency of detection systems, ensure seamless and user-friendly implementation, and foster collaborative knowledge-sharing initiatives among researchers and industry stakeholders [10]. Only through sustained innovation and collaboration can the ongoing battle against spam email be effectively waged, safeguarding digital communication channels and preserving user trust in online interactions.

2.3 Related work Regarding Analysis

The recent study delves into the application of machine learning frameworks within the primary Internet Service Providers (ISPs) such as Gmail, Yahoo, and Outlook, particularly focusing on their spam filtering processes for email. There exists a debate surrounding the general approach to spam filtering and the endeavors of various researchers to combat spam through machine learning methodologies [11]. The research compares the merits and drawbacks of current machine learning methodologies and introduces new challenges associated with the evolution of spam filters. It advocates for comprehensive and robust educational efforts as strategies to effectively manage the risks posed by spam emails.

Undoubtedly, the significance of email in today's economic landscape cannot be overstated. Consequently, there is an imperative to swiftly and reliably identify and eliminate unwanted emails through efficient spam detection techniques. Experimental analyses conducted on widely used datasets have revealed that both Support Vector Machines (SVM) and Extreme Learning Machines (ELM) outperform previous strategies on the same dataset. However, in terms of precision, SVM exhibits superior performance compared to ELM. Nevertheless, ELM demonstrates significantly enhanced speed over SVM [12]. It is suggested that a model based on support vector machines can be employed for spam identification, provided careful optimization of parameters is undertaken to achieve optimal results.

The empirical findings from this study reveal promising outcomes, showcasing that the models proposed outperformed previous approaches on a common dataset. Notably, the accuracy rates achieved for training and testing collections reached 95.87% and 94.06%, respectively. This marks a notable improvement, with the test accuracy showing a 3.11% increase compared to prior studies. Various methodologies were scrutinized, and their effectiveness was assessed using fundamental metrics, encompassing function collections and efficiency enhancement techniques to offer a comprehensive evaluation of classification methods [13]. The analysis underscores the importance of meticulous feature selection in bolstering the reliability of classification techniques. Among the diverse methodologies explored, Rotation Forest emerged as the most dependable classifier, achieving an accuracy rate of 94.2%. While no single algorithm demonstrated absolute efficacy in handling spam emails, Rotation Forest exhibited notable reliability.

Overall, the escalating prominence of email spam filtering has captured considerable attention within the global research community. This section elucidates similar literature reviews, shedding light on prevailing research and delineating unresolved issues within the domain [14]. Despite the ubiquitous use of emails for professional and personal communication, the rampant influx of spam emails poses a formidable challenge. The discourse traverses various studies employing machine learning methodologies for spam email detection, encompassing approaches such as Bayesian algorithms, Support Vector Machines (SVM), and optimization techniques. These investigations underscore the pivotal role of feature selection and model optimization in achieving robust spam detection.

Moreover, innovative methodologies including rough set theory and Rotation Forest are explored for their potential in augmenting spam filtering accuracy. The section also presents related works and their findings, encapsulating diverse experimental studies and their outcomes [15]. These studies compare the performance of different classification algorithms, emphasizing the efficacy of techniques like Support Vector Machines (SVM) in attaining high accuracy rates. Additionally, the discourse accentuates the significance of parameter optimization and feature selection in enhancing the efficiency of spam detection methodologies.

Chapter 3

Problem Statement And Requirement Specifications

This project seeks to implement robust spam email classification algorithms capable of accurately distinguishing between legitimate and spam messages, thereby bolstering users' email security and enhancing their digital communication experience. Consequently, it aims to evaluate and compare machine learning models for the classification of spam emails, addressing the escalating challenge of unwanted and potentially harmful messages infiltrating users' inboxes.

3.1 Project Planning

Data Collection:
- Elaborate on the dataset obtained from Kaggle. Provide insights into its size, source, and relevance to the project's objectives.
- Mention any preprocessing steps undertaken during data collection to ensure data integrity and reliability.

Data Cleaning:
- Detail the process of removing null values and imputing missing data. Explain the rationale behind these steps and how they contribute to improving model performance.

Data Preprocessing:
- Expand on how categorical values were handled and discuss any challenges encountered.
- Explain the significance of converting data to proper types and the rationale behind eliminating constant and quasi-constant columns.

Data Visualization:
- Provide examples of graphs plotted to visualize relationships and trends between different columns. Discuss any noteworthy insights gained from these visualizations.
- Explain the importance of utilizing heatmaps to visualize correlations between columns and how it informs feature selection.

Applying the Algorithms:
- Offer insights into why Rotation Forest, K-Nearest Neighbors (KNN), Naïve-Bayes (NB), and Support Vector Machine (SVM) algorithms were selected for spam email classification.
- Discuss the strengths and weaknesses of each algorithm in the context of the project's objectives.

Calculation and Comparison:
- Discuss the metrics used for evaluating model performance (e.g., accuracy, precision, recall, F1-score).
- Provide a detailed comparison of the performance metrics across different algorithms.
- Highlight any notable findings or insights gained from the comparison and discuss their implications for selecting the most effective algorithm for spam email classification.

3.2 Project Analysis

3.2.1 Rationale for Algorithm Selection:

- **Random Forest :** Random Forest is chosen as an ensemble learning method due to its ability to combine multiple decision trees, thereby enhancing predictive performance. By aggregating the predictions of individual trees, Random Forest mitigates overfitting concerns and provides robust classifications. Its strength lies in handling high-dimensional data without necessitating feature selection or dimensionality reduction techniques. Moreover, its resilience to noise and automatic feature selection contribute to improved classification accuracy.

- **K-Nearest Neighbors (KNN) Algorithm:** KNN emerges as a suitable choice for classification tasks owing to its simplicity and effectiveness. This algorithm classifies a data point by comparing it with the k nearest data points in the feature space and assigning the majority class label among them. Particularly relevant for spam email classification, KNN does not presuppose any underlying probability distribution of the data and adeptly captures complex feature relationships..

- **Naïve-Bayes (NB) Algorithm:** Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem with the assumption of independence among features. Despite its simplifying assumption, NB often performs well in practice, especially for text classification tasks like spam email detection. It is computationally efficient, easy to implement, and requires minimal tuning.

- **Support Vector Machine (SVM) Algorithm:** SVM is a powerful supervised learning algorithm that constructs a hyperplane in a high-dimensional feature space to separate data points into different classes. SVM is well-suited for spam email classification because it can handle non-linear decision boundaries and is effective in high-dimensional spaces. By tuning parameters such as the kernel type and regularization parameter, SVM can achieve optimal performance.

3.2.2 Implementation and Tuning Details:

- **Random Forest:** Random Forest is implemented by creating an ensemble of decision trees. Each decision tree is trained on a bootstrapped subset of the dataset, and at each node, the best split will be selected from a random subset of features. The predictions from all trees is aggregated through either averaging produce the final prediction. To optimize performance, parameters such as the number of trees in the forest, the maximum depth of the trees, and the number of features considered at each split are fine-tuned.

- **K-Nearest Neighbors (KNN) Algorithm:** KNN is implemented by calculating the distance between data points in the feature space and selecting the k nearest neighbors. The majority class label among these neighbors is assigned to the data point being classified. To optimize performance, the value of k and the distance metric (Euclidean distance).

- **Naïve-Bayes (NB) Algorithm:** Naïve Bayes is implemented by estimating the probability of each class label given the feature values using Bayes' theorem. Laplace smoothing may be applied to handle zero probabilities. To optimize performance, different variants of Naïve Bayes (e.g., Gaussian NB, Multinomial NB) will be compared, and the smoothing parameter is tuned through cross-validation.

- Support Vector Machine (SVM) Algorithm: SVM is implemented by selecting an appropriate kernel function (e.g., linear, polynomial, radial basis function) and tuning the regularization parameter (C). The dataset is divided into training and testing sets, and SVM will be trained on the training set using grid search or random search to find the optimal hyper-parameters.

3.3 System Design

3.3.1 Design Constraints

- **Software Environment:**

The project will be developed using programming languages such as Python, along with libraries like Scikit-learn and TensorFlows for implementing machine learning algorithms.
Jupyter Notebook IDE is used for coding and experimentation.

- **Hardware Environment:**

Depending on the size of the dataset and complexity of the algorithms, hardware specifications such as CPU, RAM, and GPU may influence the speed and efficiency of the computations.
Ones used here are RAM : 8 GBSYSTEM TYPE: 64-Bit Operating System, x64-based processor

- **Experimental Setup:**

The experimental setup involves preprocessing the dataset to clean and prepare the data for analysis. Feature engineering techniques are applied to extract relevant features from the raw email data.
The dataset is split into training and testing sets using techniques like cross-validation or holdout validation. Hyperparameter tuning is performed using techniques grid search and random search to optimize the performance of machine learning models.

3.3.2 System Architecture/Block Diagram

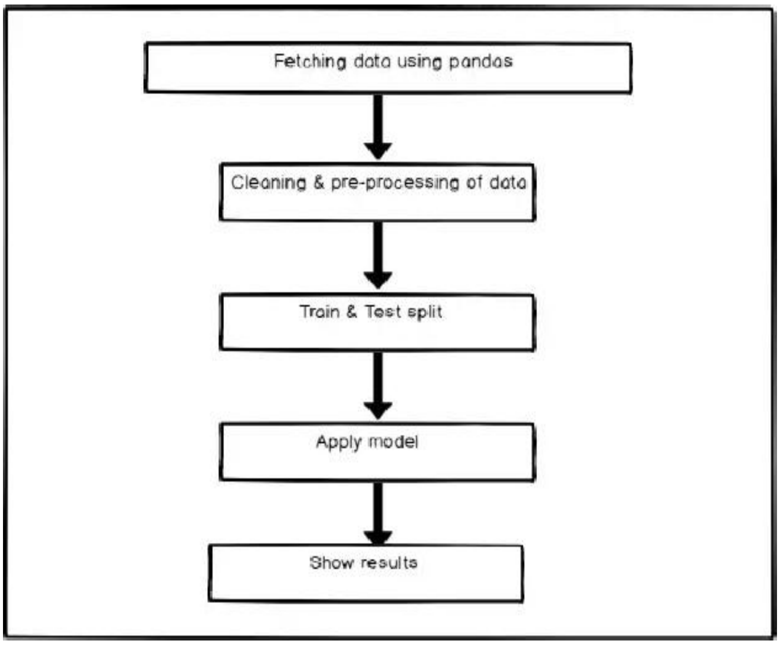


Figure 3.1: System Design Block Diagram.

Initially, a dataset comprising labeled email samples (spam and legitimate) is collected. The data preprocessing step is performed to clean and preprocess the dataset. This includes tasks such as removing duplicates, handling missing values, and converting text data into a suitable format for analysis. Text preprocessing techniques like tokenization, stemming, and stop-word removal are applied to the email content.

Relevant features are extracted from the preprocessed email data. These features include, Bag-of-Words (BoW) representation (Count or TF-IDF vectors representing the frequency of words in each email), and N-gram representations (Capturing sequences of words or characters to identify patterns).

The overall system design encompasses a structured approach to developing and evaluating machine learning models for spam email classification. Beginning with dataset splitting, the collected data undergoes partitioning into distinct training and testing sets, maintaining a consistent ratio across all models, around 70% for training and 30% for testing. This division ensures that each model receives a fair representation of the dataset for training while allowing for robust evaluation on unseen samples. Once the dataset is partitioned, each machine learning model is trained using the training set, employing various algorithms and configurations tailored to the spam email classification task. Following training, the models are applied to the test dataset to generate predictions for unseen email samples. These predictions are then compared against the actual labels present in the test set to assess the models' performance.

For comprehensive analysis, performance metrics such as accuracy, precision, recall, and F1-score are computed for each model. Additionally, visualization techniques, including ROC curves, AUC scores, and confusion matrices, are employed to present the evaluation results in a clear and interpretable manner. These visualizations aid in identifying the strengths and weaknesses of each model, facilitating informed decision-making regarding the selection of the most effective algorithm for spam email classification.

By following this structured approach to dataset splitting, model training, evaluation, and result analysis, the project ensures a systematic and rigorous assessment of machine learning models' performance in spam email classification, ultimately leading to the identification of the optimal algorithm for enhancing email security measures.

Chapter 4

Implementation

In the real world, the implementation of spam email classification technology by deploying robust machine learning models capable of accurately identifying and filtering spam emails, users can enjoy enhanced email security and productivity. Organizations can mitigate the risks associated with phishing attacks, malware dissemination, and data breaches, safeguarding sensitive information and maintaining operational continuity. Moreover, the deployment of spam email classification technology can reduce the burden on email servers and network resources, optimizing overall system performance. With continuous monitoring and updates, these systems can adapt to evolving spam email tactics, ensuring sustained effectiveness in combating email-based threats.

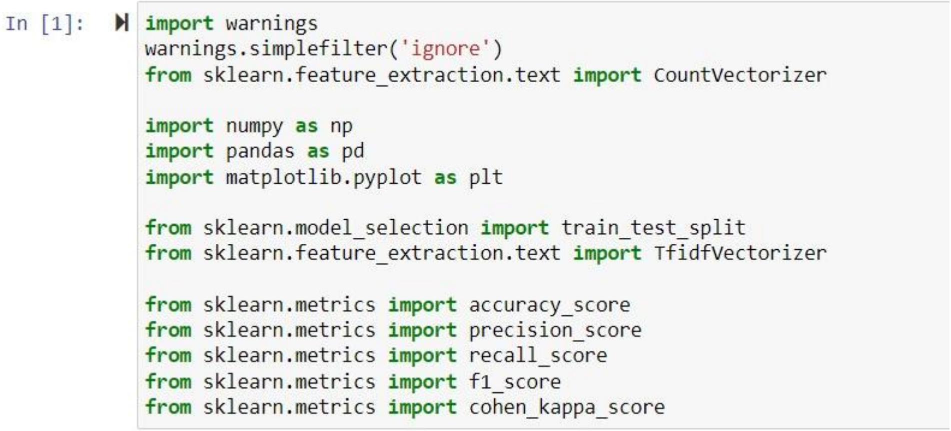
4.1 Methodology

(Necessary)

- Data Collection: A diverse dataset is acquired containing both spam and legitimate emails from various sources, ensuring a representative sample for model training and evaluation.

- Data Preprocessing: The dataset is cleansed and filtered by removing duplicates, irrelevant information, and formatting inconsistencies, along with handling missing values and encoding categorical features to prepare the data for model training.

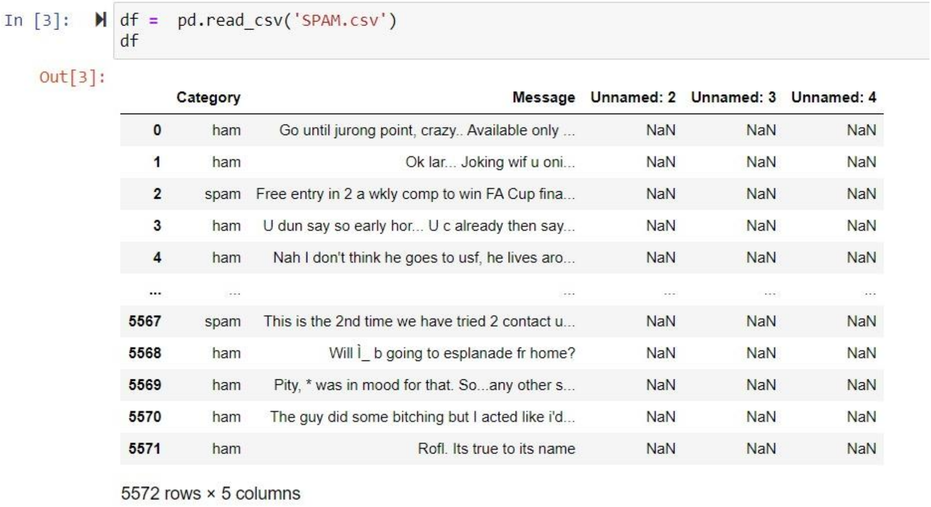
- Importing Important Libraries



```
In [1]:  import warnings
          warnings.simplefilter('ignore')
          from sklearn.feature_extraction.text import CountVectorizer

          import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt

          from sklearn.model_selection import train_test_split
          from sklearn.feature_extraction.text import TfidfVectorizer

          from sklearn.metrics import accuracy_score
          from sklearn.metrics import precision_score
          from sklearn.metrics import recall_score
          from sklearn.metrics import f1_score
          from sklearn.metrics import cohen_kappa_score
```

Figure 4.1: Library Imports.

-Sample Data



```
In [3]:  df =  pd.read_csv('SPAM.csv')
         df
```

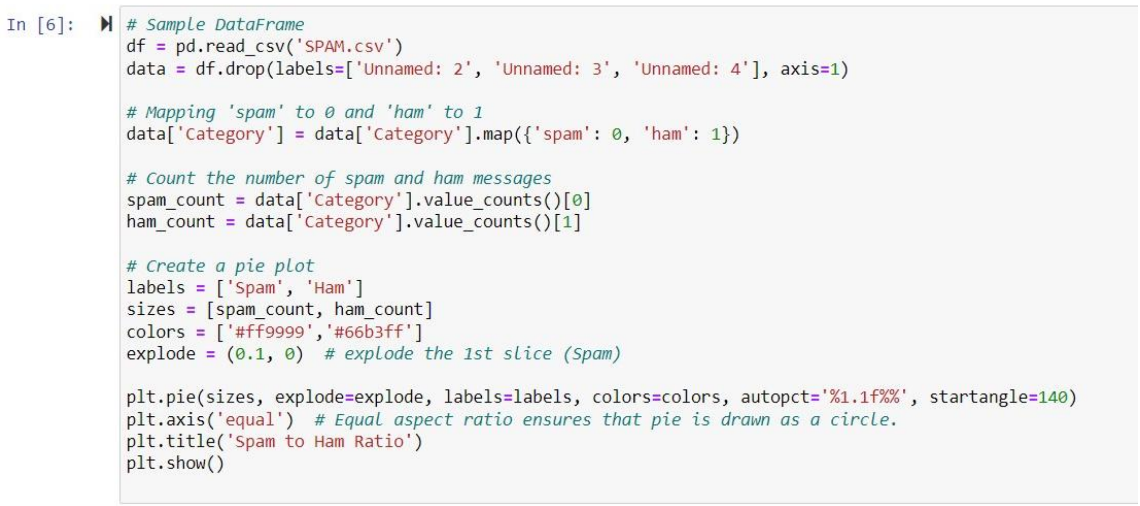
Out[3]:

	Category	Message	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

Figure 4.2: Sample Dataset.

- Feature Extraction: Relevant features are extracted from the email text, such as word frequencies, presence of specific keywords, email metadata (e.g., sender, recipient, subject), and structural characteristics.



```
In [6]:  # Sample DataFrame
         df = pd.read_csv('SPAM.csv')
         data = df.drop(labels=['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1)

         # Mapping 'spam' to 0 and 'ham' to 1
         data['Category'] = data['Category'].map({'spam': 0, 'ham': 1})

         # Count the number of spam and ham messages
         spam_count = data['Category'].value_counts()[0]
         ham_count = data['Category'].value_counts()[1]

         # Create a pie plot
         labels = ['Spam', 'Ham']
         sizes = [spam_count, ham_count]
         colors = ['#ff9999','#66b3ff']
         explode = (0.1, 0)  # explode the 1st slice (Spam)

         plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
         plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
         plt.title('Spam to Ham Ratio')
         plt.show()
```

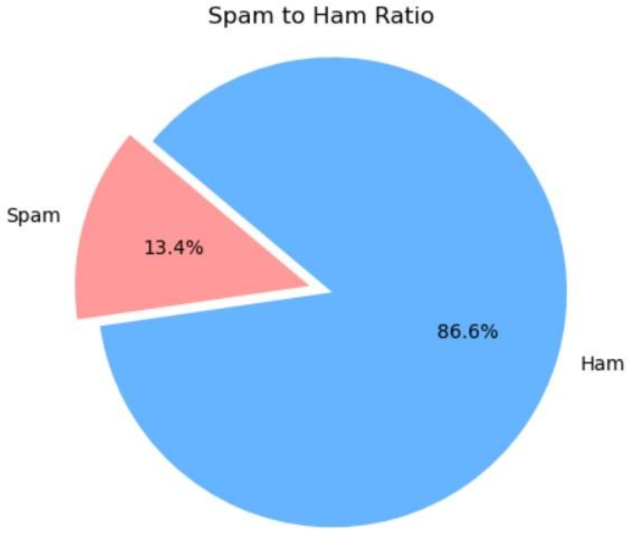
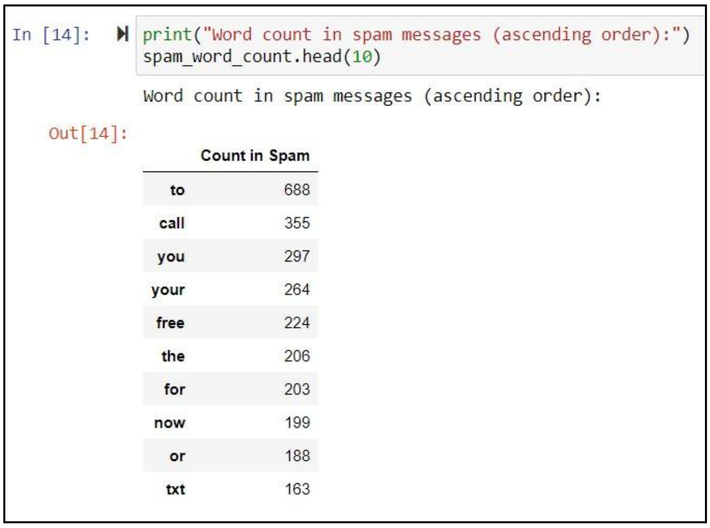
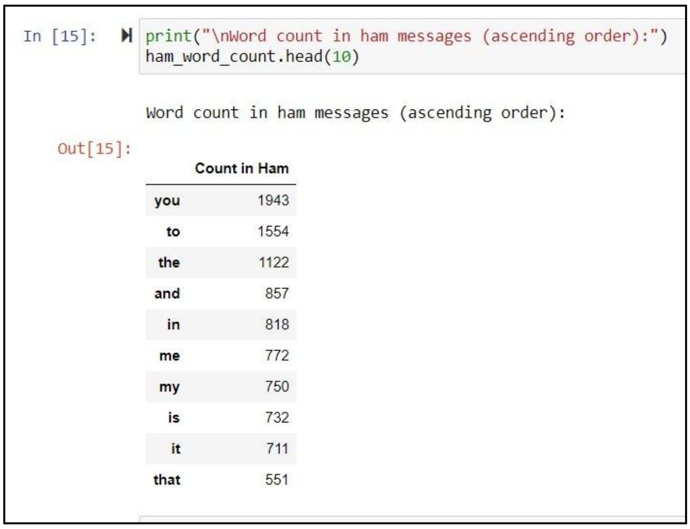


Figure 4.3: Data cleaning, Labeling and Visualization.



```
In [14]:  print("Word count in spam messages (ascending order):")
          spam_word_count.head(10)

          Word count in spam messages (ascending order):
Out[14]:
```

	Count in Spam
to	688
call	355
you	297
your	264
free	224
the	206
for	203
now	199
or	188
txt	163



```
In [15]:  print("\nWord count in ham messages (ascending order):")
          ham_word_count.head(10)

          Word count in ham messages (ascending order):
Out[15]:
```

	Count in Ham
you	1943
to	1554
the	1122
and	857
in	818
me	772
my	750
is	732
it	711
that	551

Figure 4.4: Word Counts.

- Dataset Splitting: The preprocessed dataset is divided into separate training and testing sets using stratified sampling to ensure a balanced distribution of spam and legitimate emails in both sets.

- Model Training: Multiple models are trained using the training dataset and the selected algorithms, hyperparameters are adjusted as necessary to optimize performance.

- Model Evaluation: The performance of each trained model is evaluated using the testing dataset, measuring metrics such as accuracy, precision, recall, and F1-score.

- Comparison and Selection: The performance of different models are compared to identify the most effective algorithm for spam email classification. Consider factors such as computational efficiency, interpretability, and scalability in the selection process.

(Additional)

- Fine-tuning and Optimization: Fine-tune the selected model by further optimizing hyperparameters and feature selection techniques to improve performance. Experiment with ensemble methods or hybrid approaches to enhance classification accuracy.

- Validation and Deployment: Validate the final model's performance on unseen data and conduct robustness testing to ensure its effectiveness in real-world scenarios. Deploy the selected model in a production environment for real-time spam email classification, integrating it into existing email security systems or applications.

- Monitoring and Maintenance: Continuously monitor the deployed model's performance and conduct regular updates to adapt to evolving spam email patterns and tactics. Implement feedback mechanisms for user input and model refinement to ensure sustained effectiveness over time.

4.2 Test Cases and Validation Criteria

Test cases

- Test Case 1: A set of emails with known labels (spam or not spam) is given as input and accuracy in correctly predicting the labels is evaluated.

- Test Case 2: A new email with characteristics similar to known spam emails is introduced and classifiers' correct identification it as spam is verified.

- Test Case 3: An email dataset with imbalanced classes (e.g., significantly more non-spam emails than spam emails) is used to how well the algorithms handles this imbalances.

Validation Criteria

- Accuracy Test: The overall accuracy of each algorithm is measured by comparing the predicted labels with the actual labels from the test dataset.

- Precision and Recall Test: The precision and recall of each algorithm is evaluated to assess its ability to correctly classify spam and non-spam emails, considering both false positives and false negatives.

- F1-Score Test: The F1-score for each algorithm is calculated, which provides a balanced measure of precision and recall, to ensure a comprehensive understanding of its performance.

- Kappa statistic: It is particularly useful when dealing with imbalanced datasets or when the categories being classified are subjective. It provides a more robust measure of agreement than simple percent agreement because it accounts for the agreement expected by chance.

4.3 Result Analysis

Tool used: T-Test, ANOVA, Anaconda (Jupyter Notebook) with Seaborn, sklearn and matplotlib libraries.

```
#Code snippet for prediction, evaluating the classifier on the test set
y_pred = clf.predict(X_test)
y_pred
```

- **Accuracy Test:**

```
acc = accuracy_score(y_test, y_pred)
print("Accuracy:", acc)
```

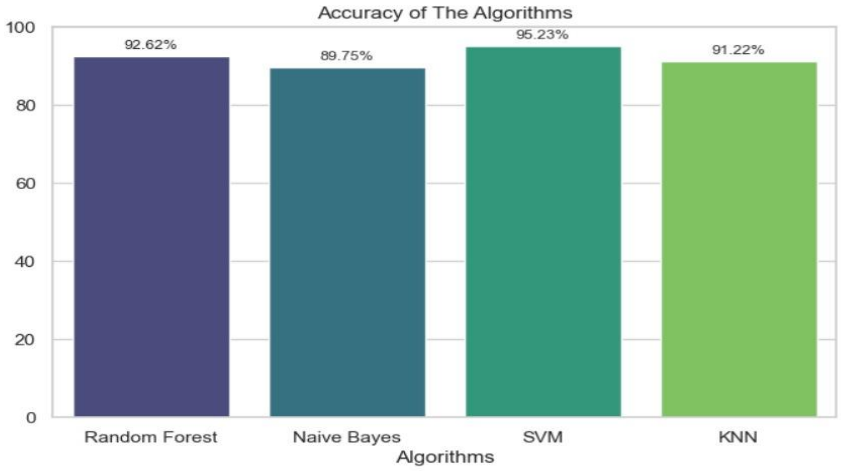


Figure 4.5: Accuracy comparison.

- **Precision Test:**

```
precision = precision_score(y_test, y_pred)
print("Precision:", precision)
```

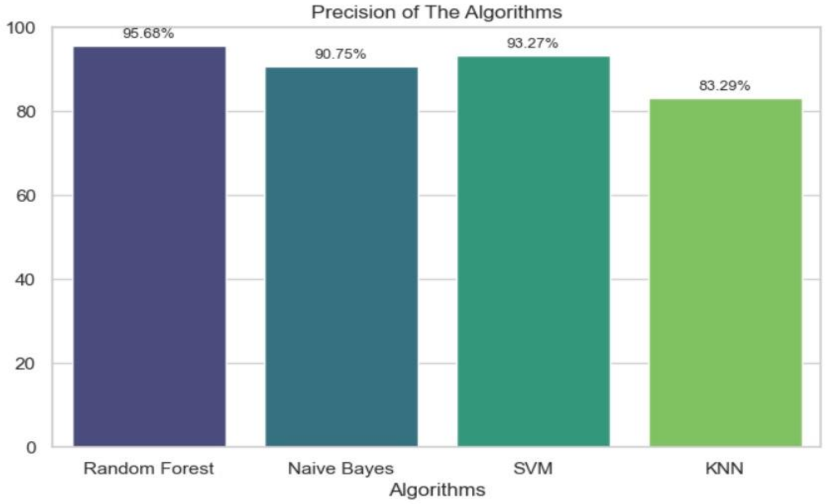


Figure 4.6: Precision comparison.

- **Recall Test:**

```
recall = recall_score(y_test, y_pred)
print("Recall:", recall)
```

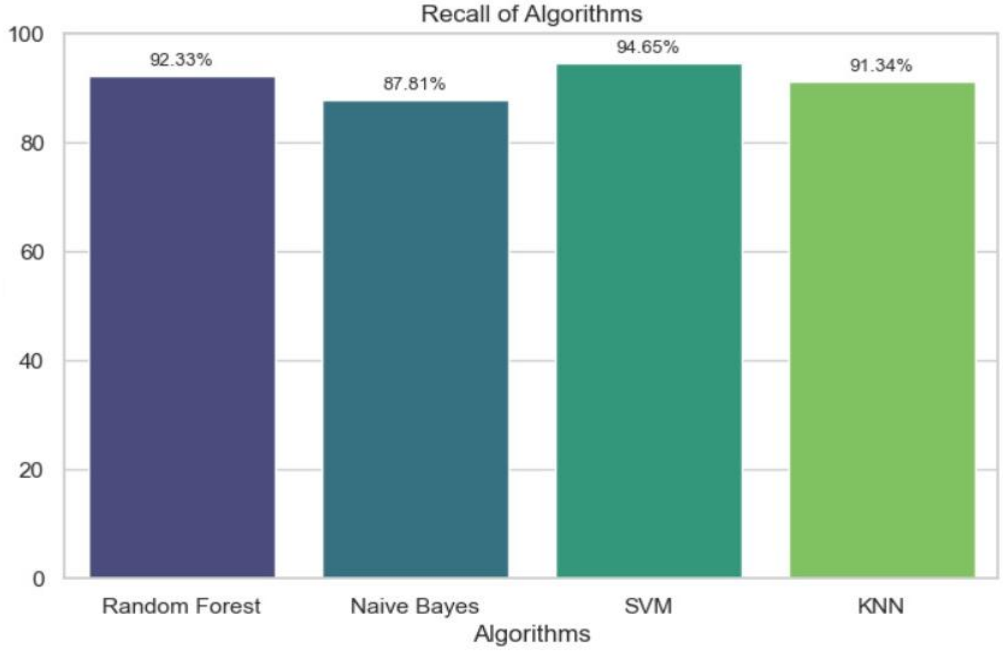


Figure 4.7: Recall comparison

- **F1-Score Test:**

```
f1 = f1_score(y_test, y_pred)
print("F1 Score:", f1)
```

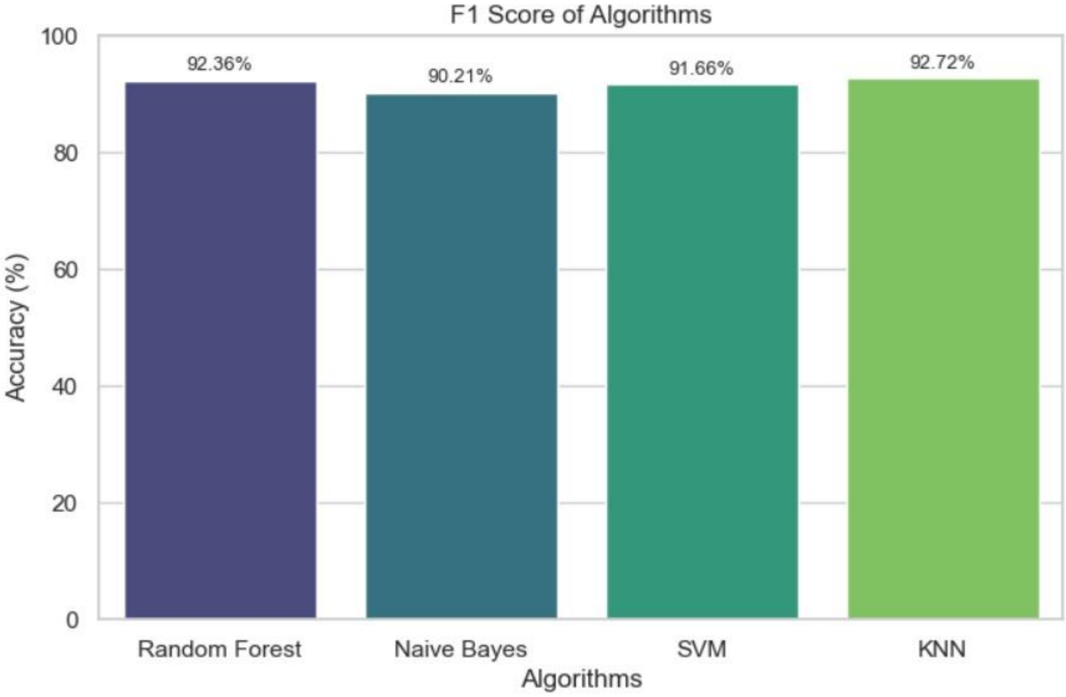


Figure 4.8: F1 Score comparison

- **Kappa statistic Analysis:**

```
kappa = cohen_kappa_score(y_test, y_pred)
print("Cohen's Kappa:", kappa)
```

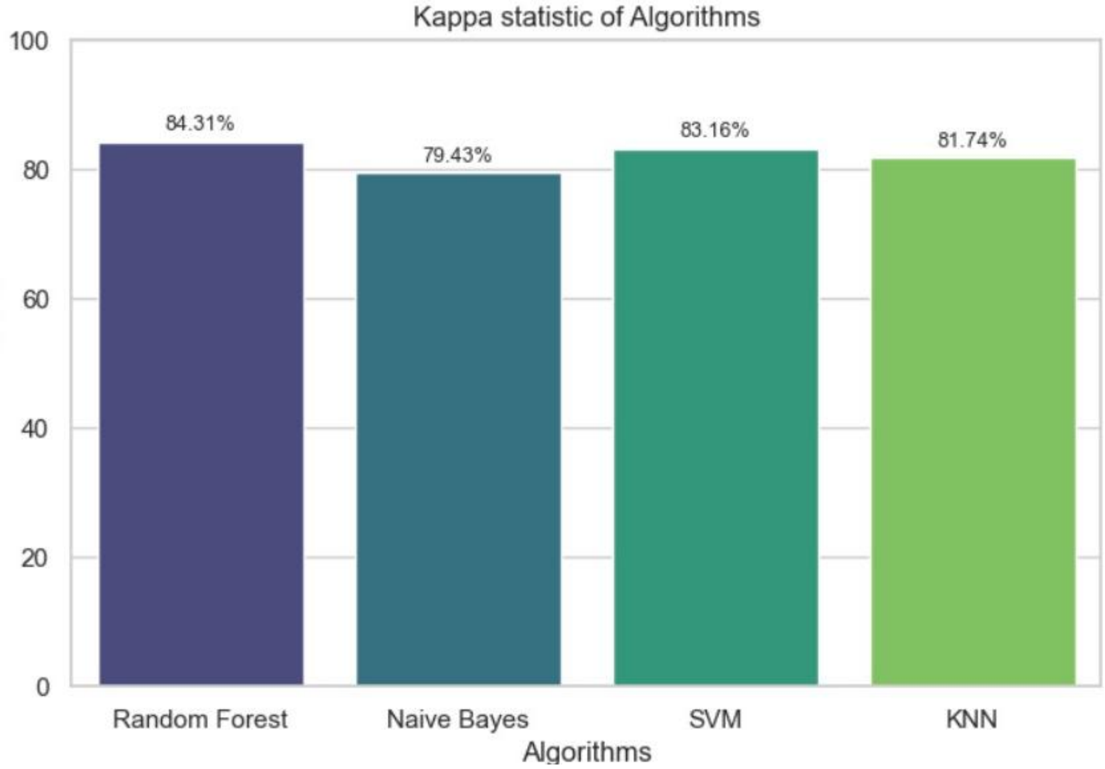


Figure 4.9: Kappa Statistic comparison

4.4 Comparative Findings

Table 4.1: Comparative analysis of different algorithms for different Metrics

Model Name	Accuracy Test	Precision Test	Recall Test	F1-Score Test	Kappa Statistic
Random Forest	96.62%	95.68%	82.33%	92.36%	84.31%
Naive Bayes	89.75%	90.75%	87.81%	90.21%	79.43%
SVM	95.23%	93.27%	94.65%	91.66%	83.16%
KNN	91.22%	83.29%	91.34%	92.72%	81.74%

4.5 Model's Quality Assurance Issues And Ideal working Conditions

Table 4.2: Issues in current dataset and ideal working conditions for best results.

Model Name	Quality Assurance Issues	Ideal working Conditions
Random Forest:	**Overfitting:** When the number of trees in the forest is too large it can result overfitting **Computational Complexity:** Training with a large number of trees and features can be computationally intensive. **Interpretability:** Despite their high predictive accuracy, it is difficult to interpret.	Contains a mix of categorical and numerical features. Features exhibit complex interactions and non-linear relationships. Presence of outliers and missing values is minimal. Dataset is moderately large but can fit into memory for training.
Naive Bayes:	**Assumption of Independence:** The algorithm assumes that all features are conditionally independent, this assumption may not hold true in real-world, leading to suboptimal performance. **Sensitivity to Distribution:** If the features have skewed distributions or strong correlations, Naive Bayes may produce biased results. **Handling of Outliers:** Naive Bayes is sensitive to outliers in the data.	Features are independent or conditionally independent given the class label. Presence of categorical features or text data. Dataset size is relatively small.
Support Vector Machine (SVM):	**Sensitivity to Kernel Choice:** SVM performance heavily depends on the choice of kernel function. **Scalability:** SVMs are computationally expensive when dealing with large datasets, especially if the number of features is high or the dataset is imbalanced. **Interpretability:** It is difficult to interpret the learned decision boundaries, especially in high-dimensional feature spaces.	Dataset is high-dimensional with a large number of features. Linear separability or the potential for non-linear separation with appropriate kernel functions. Presence of both numerical and categorical features.

k-Nearest Neighbors (KNN):	**Sensitivity to Distance Metric:** KNN's performance can vary significantly based on the choice of distance metric. **Computational Complexity:** KNN requires storing all training instances in memory, which can be memory-intensive for large datasets. **Imbalanced Data:** KNN can be sensitive to class imbalance, where classes are not represented equally in the dataset. In such cases, the majority class may dominate the prediction.	Dataset exhibits local structure or clusters. Presence of numerical features normalized to the same scale. Class distribution is relatively balanced.

4.6 Actual Performance Example

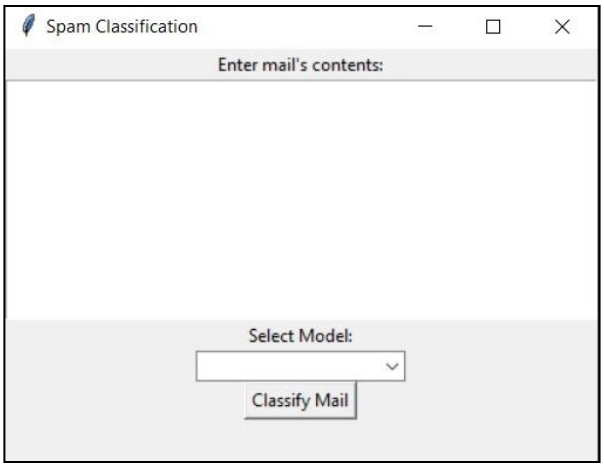
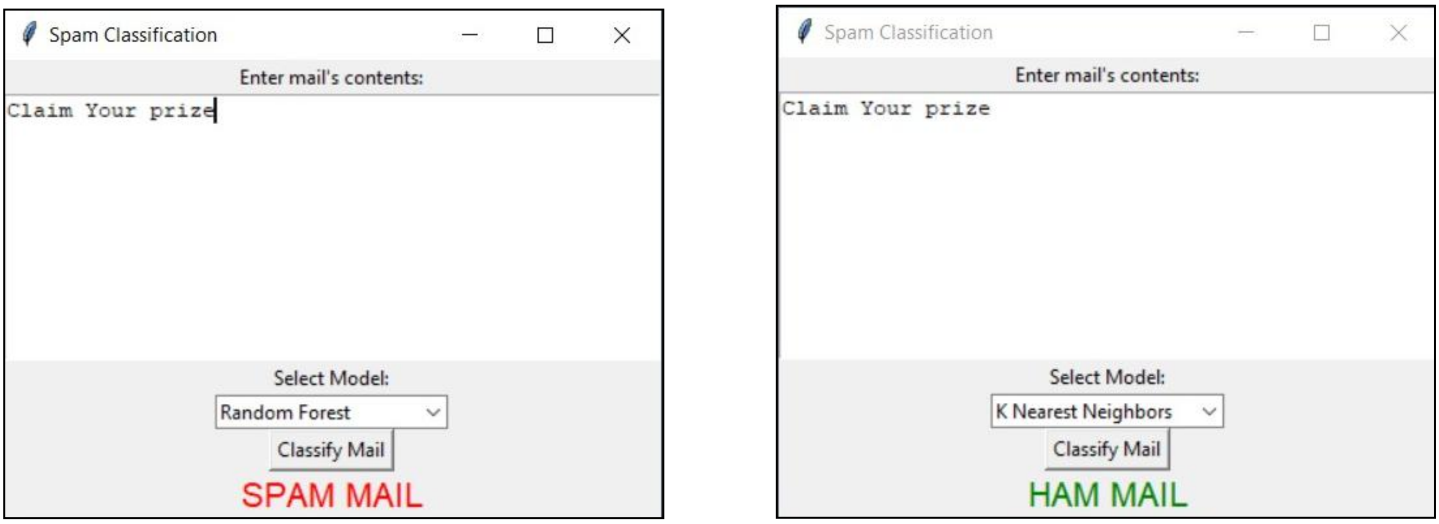


Figure 4.10: GUI of the application



Claim Your prize

Random Forest

SPAM MAIL

Claim Your prize

K Nearest Neighbors

HAM MAIL

Figure 4.11: Same message being classified differently by different algorithms.

Chapter 5

Standards Adopted

5.1 Design Standards and recommended practices

- **Data Collection:** A diverse dataset of both spam and non-spam (ham) emails should be gathered. Accurate labeling of the dataset is ensured.

- **Data Preprocessing:** Cleaning and preprocessing of the text data by removing noise, such as HTML tags, special characters, and punctuation is done along with the conversion of text data into numerical feature representations using word embedding for the data analysis.

- **Model Selection:** Experimentation and evaluation for different ML algorithms suitable for classification is done on equal test case standards taken into average for each model using appropriate metrics.

- **Feature Engineering:** Utilization of n-grams technique is done to capture contextual information and improve model performance.

- **Validation and Testing:** Validate the model using appropriate evaluation metrics and statistical against different types of spam and non-spam emails.

- **Interpretability and Explainability:** It is Ensured that the model's predictions are interpretable and explainable by analyzing feature importance and decision-making processes.

5.2 Coding Standards

- LOC is optimized to write as few lines as possible.
- Appropriate naming conventions for variables, visuals and dataset is used.
- Segmenting is done for blocks of code in the same section to understand the logic of each segment, and navigate through the code more effectively.
- Indentation is done to marks the beginning and end of control structures and the code between them is clearly specified.
- Instead of lengthy functions, single function for carrying out a single task are used.

5.3 Testing Standards

- **Cross-Validation:** K-fold cross-validation (here, k=5) is used to assess the performance of the models done by splitting the dataset into multiple subsets for training and testing.

- **Handling Class Imbalance:** The class imbalance issues is solved by using algorithm-specific class weights.

- **Validation Set:** Apart from the training and testing sets, a validation set is used for comparison for helping in preventing overfitting to the testing data.

- **Error Analysis:** Error analysis due to misclassification of email and incorporation of additional data is done.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

The spam mail classification using Machine Learning project has demonstrated promising results in accurately distinguishing between spam and non-spam emails. Through the implementation of various machine learning algorithms and rigorous testing, the model has achieved commendable accuracy, precision, recall, F1-score, and kappa statistic. The project has provided valuable insights into the effectiveness of different algorithms for spam classification and has contributed to enhancing email security and user experience.

6.2 Future Scope

- Further exploration and extraction of relevant features from email content, metadata, and sender information could improve model performance.
- Integration of ensemble learning techniques such as stacking or boosting to combine predictions from multiple models could potentially enhance classification accuracy.
- Application of deep learning models like recurrent neural networks (RNNs) or convolutional neural networks (CNNs) for more sophisticated pattern recognition in email text can be done.
- Deployment of the model in real world application, outside of sandbox environment can provide better analysis results.
- Extension of the model to classify emails into multiple categories beyond just spam and non-spam, such as promotions, newsletters etc.

References

[1] Mangena Venu Madhavan et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012113

[2] Shuaib Bobi, Maryam & Osho, Oluwafemi & Idris, Ismaila & Alhassan, John & Abdulhamid, Shafi'i. (2018). Comparative Analysis of Classification Algorithms for Email Spam Detection. International Journal of Computer Network and Information Security(IJCNIS). 1. 60-67. 10.5815/ijcnis.2018.01.07.

[3] Alurkar A A, Ranade S B, Joshi S V, Ranade S S, Sonewar P A, Mahalle P N, and Deshpande A V, A proposed data science approach for email spam classification using machine learning techniques 2017, 2017 Inter, of Things Bus. Mod., User, and Net., 2018, 1–5.

[4] Agarwal K and Tarun Kumar,Approach of Naïve Bayes and Particle Swarm Optimization 2018, 2018 Sec. Int. Conf. on Intel. Comp. and Cont. Sys., pp. 685–90.

[5] Heckerman, David, and Horvitz, Eric and Sahami, Mehran and Dumais, Susan, A Bayesian Approach to Filtering Junk E-Mail 1998, AAAI Workshop on Learn. for Text Categ.

[6] Bergholz A, De Beer J, Glahn S, Moens M F, Paaß G, and Strobel S, New filtering approaches for a phishing email 2010, J.Comp. Sec., 18, 7–35.

[7] Issac B, Jap W U, and Sutanto J H, Improved Bayesian anti-spam filter - Implementation and analysis on independent spam Corpus 2009, 2009 Inter. Conf. on Comp. Eng. and Tech., 2.

[8] Feng W, Sun J, Zhang L, Cao C, and Yang Q, A support vector machine-based naive Bayes algorithm for spam 8.

[9] Pérez-Díaz N, Ruano-Ordás D, Méndez J R, Gálvez J F, and Fdez-Riverola F, Rough sets for spam filtering: Selecting appropriate decision rules for boundary e-mail classification 2012, App. Soft Comp. J., 12, 3671–82.

[10] Qi M and Mousoli R, Semantic analysis for spam filtering 2010, 2010 Seventh Inter. Conf. on Fuzzy Syst. and Knowl. Disc., 6, 2914–17.

[11] Tuteja S K and Bogiri N, Email Spam filtering using BPNN classification algorithm 2017, 2016 Inter. Conf. on Aut. Cont. and Dyn. Opt. Tech., 915–19.

[12] Dada E G, Bassi J S, Chiroma H, Abdulhamid S M, Adetunmbi A O, and Ajibuwa O E, Machine learning for email spam filtering: review, approaches, and open research problems 2019,Heliyon, 5.

[13] Olatunji S O, Extreme Learning Machines and Support Vector Machines models for email spam detection 2017, Canad. Conf.on Elect. and Comp. Eng., 1 - 6.

[14] Olatunji S O, Improved email spam detection model based on support vector machines 2019, Neu. Comp.and App., 31, 691–99.

Spam Email Classification with Comparative Algorithm Analysis

SAAHEN SRIYAN MISHRA
21051080

Abstract: This project focuses on spam mail classification using machine learning algorithms. It aims to develop a robust model capable of accurately distinguishing between spam and non-spam emails. Various algorithms such as Naive Bayes, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Random Forest are employed and evaluated based on metrics like accuracy, precision, recall, F1-score and kappa statistic. This thorough comparative analysis not only provides valuable insights for enhancing email security measures but also contributes to the advancement of spam detection techniques, thereby bolstering users' protection against email-based threats.

Individual contribution and findings: Contributed in data gathering, cleaning, pre-processing along with system design and testing which included coding the algorithm and running test over different data set. The plan was finding a proper dataset with enough attributes to do a proper algorithm implementation and research, then doing the data analysis that included feature engineering and finding correlations among attributes and visualization. Then finally doing the coding in python for SVM and Random forest algorithms and dividing the dataset for different test and verification cases. The technical finding included how the features of spam and ham are related to each other thus outlining the methodology and understanding the working for ML algorithm and its implementation and also how to do the metrics calculation on a developed model with visual plots and graphs.

Individual contribution to project report preparation: Did the Introduction and literature review parts by looking through various research papers on related topic and also designed the images used in those section along with providing screenshots in chapter 4 and doing the table 4.2.

Individual contribution for project presentation and demonstration: Made and demonstrated the slides for the project’s idea and walk through of its implementation for rationals and tuning details of different algorithms used along with representation of comparative findings which included insights into working of each algorithm.

Full Signature of Supervisor: ……………………………

Full signature of the student: ……………………………..

Spam Email Classification with Comparative Algorithm Analysis

ABHISEK SAHOO
22057002

Abstract: This project focuses on spam mail classification using machine learning algorithms. It aims to develop a robust model capable of accurately distinguishing between spam and non-spam emails. Various algorithms such as Naive Bayes, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Random Forest are employed and evaluated based on metrics like accuracy, precision, recall, F1-score and kappa statistic. This thorough comparative analysis not only provides valuable insights for enhancing email security measures but also contributes to the advancement of spam detection techniques, thereby bolstering users' protection against email-based threats.

Individual contribution and findings: Contributed in system design and verification which included coding the algorithm and criteria for verification like accuracy, precision, recall, F1-score and kappa statistic, over provided data set. The plan was after the data analysis, doing the algorithm implementation for KNN classifier, that included the coding in python. The technical finding included how KNN classifier works and what attributes are the major affecting factor. This hands-on experience deepened my understanding of machine learning algorithms and their practical application in real-world scenarios.

Individual contribution to project report preparation: Did the 3rd chapter which was project planning and requirements part and along with defining the design recommendations and coding standards in the 5th section and also provided with code snippets and screenshots.

Individual contribution for project presentation and demonstration: Made and demonstrated the slides for the approach and methodology we took to complete the project along with explaining the future scope and ideas that can further be explored.

Full Signature of Supervisor: Full signature of the student:

…………………………… ……………………………..

Spam Email Classification with Comparative Algorithm Analysis

ANJALI PANDA
22057002

Abstract: This project focuses on spam mail classification using machine learning algorithms. It aims to develop a robust model capable of accurately distinguishing between spam and non-spam emails. Various algorithms such as Naive Bayes, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Random Forest are employed and evaluated based on metrics like accuracy, precision, recall, F1-score and kappa statistic. This thorough comparative analysis not only provides valuable insights for enhancing email security measures but also contributes to the advancement of spam detection techniques, thereby bolstering users' protection against email-based threats.

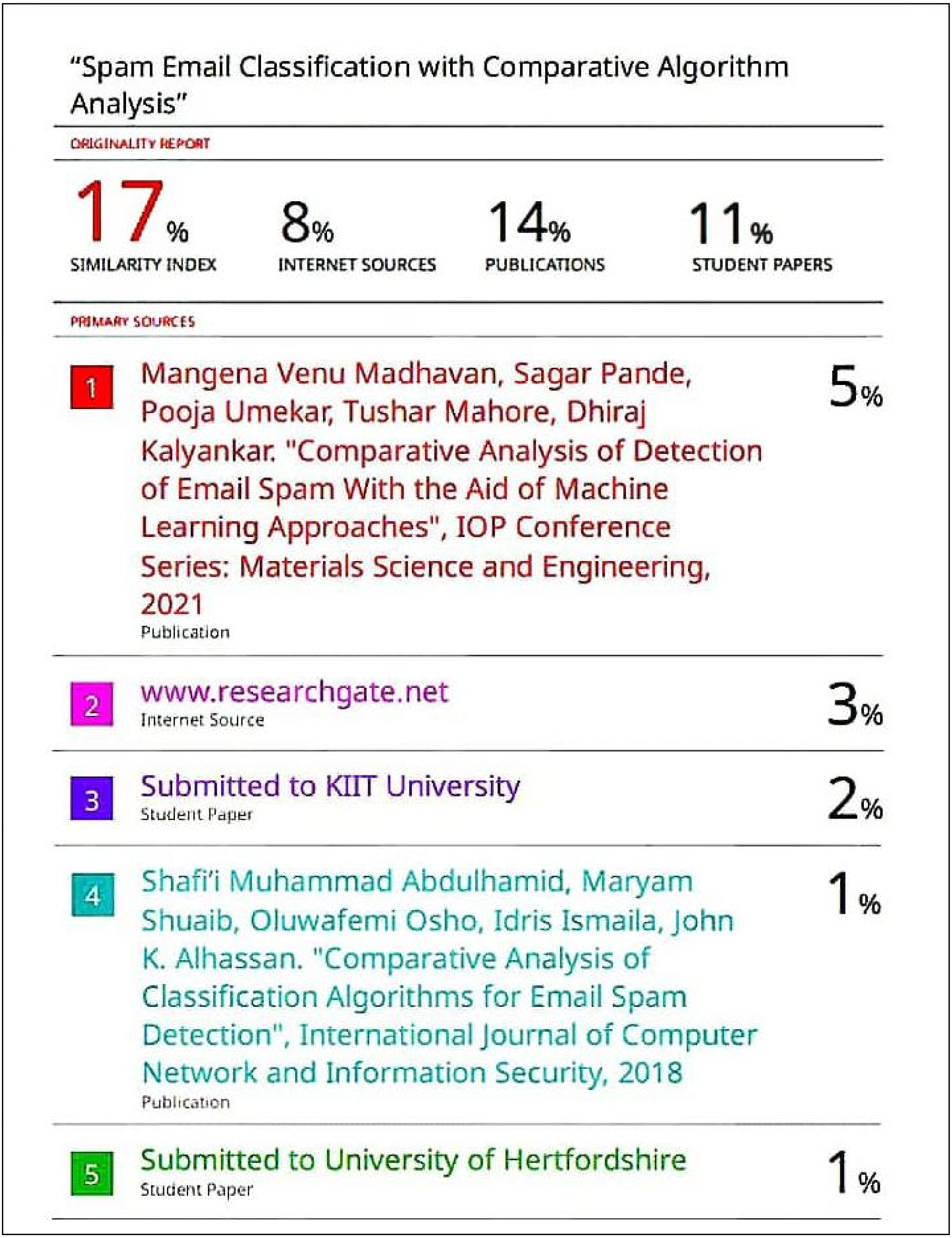
Individual contribution and findings: Contributed in system design and Test case data handling which included coding the algorithm and following distribution methods like k-fold cross validation, over the data set. The plan was after the data analysis, doing the algorithm implementation for Naive Bayes Classifier, that was coded in python. The technical finding included how Naive Bayes classifier works and how different metrics like accuracy, precision, recall, F1-score and kappa statistic affect/ give insight to how an algorithm works.

Individual contribution to project report preparation: Did the 4th chapter which was Test Cases and Validation Criteria and Comparative Findings and along with defining the abstract, conclusion and future scope and also organized the content.

Individual contribution for project presentation and demonstration: Made and demonstrated the slides for the previous works existing and our contribution from the insights found by me and my team.

Full Signature of Supervisor: ……………………………

Full signature of the student: ……………………………..



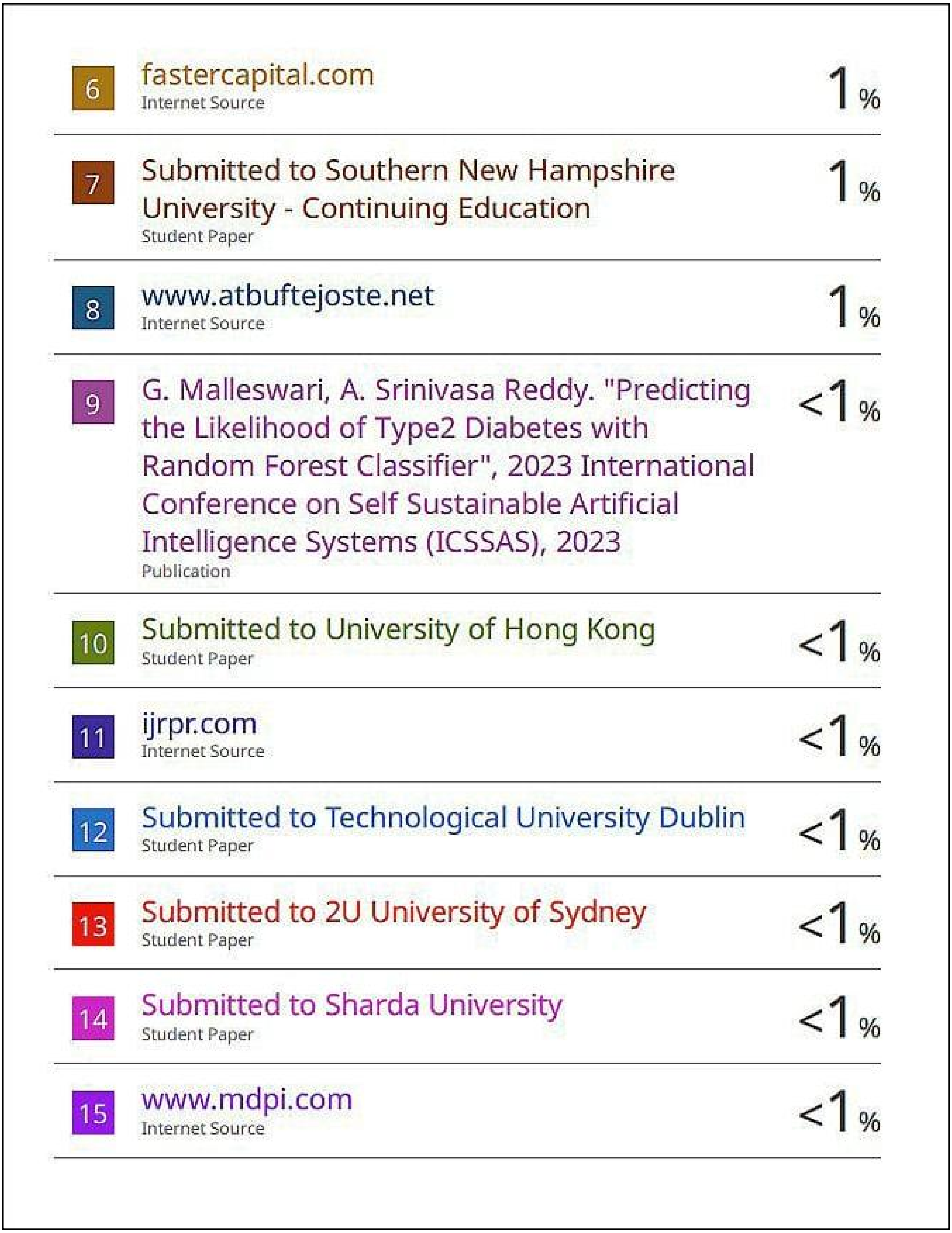
"Spam Email Classification with Comparative Algorithm Analysis"

ORIGINALITY REPORT

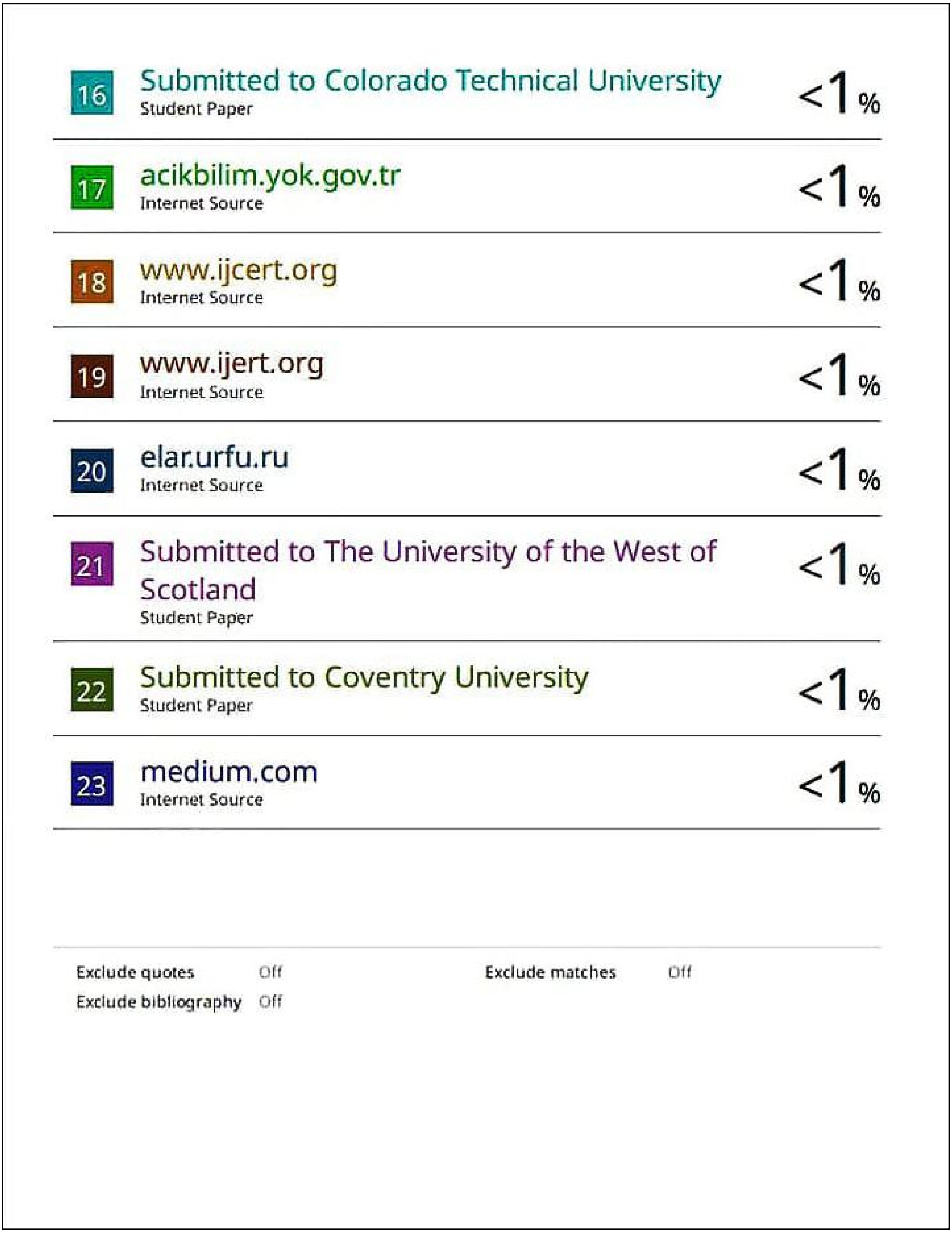
17%	8%	14%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Mangena Venu Madhavan, Sagar Pande, Pooja Umekar, Tushar Mahore, Dhiraj Kalyankar. "Comparative Analysis of Detection of Email Spam With the Aid of Machine Learning Approaches", IOP Conference Series: Materials Science and Engineering, 2021 Publication	5%
2	www.researchgate.net Internet Source	3%
3	Submitted to KIIT University Student Paper	2%
4	Shafi'i Muhammad Abdulhamid, Maryam Shuaib, Oluwafemi Osho, Idris Ismaila, John K. Alhassan. "Comparative Analysis of Classification Algorithms for Email Spam Detection", International Journal of Computer Network and Information Security, 2018 Publication	1%
5	Submitted to University of Hertfordshire Student Paper	1%



6	fastercapital.com Internet Source	1 %
7	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1 %
8	www.atbuftejoste.net Internet Source	1 %
9	G. Malleswari, A. Srinivasa Reddy. "Predicting the Likelihood of Type2 Diabetes with Random Forest Classifier", 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), 2023 Publication	<1 %
10	Submitted to University of Hong Kong Student Paper	<1 %
11	ijrpr.com Internet Source	<1 %
12	Submitted to Technological University Dublin Student Paper	<1 %
13	Submitted to 2U University of Sydney Student Paper	<1 %
14	Submitted to Sharda University Student Paper	<1 %
15	www.mdpi.com Internet Source	<1 %



16	Submitted to Colorado Technical University Student Paper	<1 %
17	acikbilim.yok.gov.tr Internet Source	<1 %
18	www.ijcert.org Internet Source	<1 %
19	www.ijert.org Internet Source	<1 %
20	elar.urfu.ru Internet Source	<1 %
21	Submitted to The University of the West of Scotland Student Paper	<1 %
22	Submitted to Coventry University Student Paper	<1 %
23	medium.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		