Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology
Kirtipur, Kathmandu
2024

Seminar Report on
"Spam Email Detection Using Support Vector Machine Technique"

In partial fulfillment of the requirement for a Master's degree in Computer Science and Information Technology (M.Sc. CSIT), 1st Semester

Submitted to:
Central Department of Computer Science and Information Technology, Tribhuvan University, Kirtipur, Kathmandu, Nepal

Submitted By:
Sijan Poudel (8015040)

Tribhuvan University
Institute of Science and Technology

Supervisor Recommendation

This is to certify that Mr. Sijan Poudel has submitted the seminar report on the topic " **Spam Email Detection Using Support Vector Machine Technique**" for the partial fulfillment of the Master of Science in Computer Science and Information Technology, First semester. I hereby declare that this seminar report has been approved.

_____

Supervisor

Assistant Professor Mr. Bikash Balami

Central Department of Computer Science and Information Technology (CD.CSIT)

Certificate of Approval

This is to certify that the seminar report prepared by Mr. Sagar Timalsena "**Spam Email Detection Using Support Vector Machine Technique"** in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

……..……..

Asst. Prof. Sarbin Sayami

(H.O.D)

Central Department of Computer Science and Information Technology

……...……

Assoc. Prof. Bikash Balami

(Supervisor)

Central Department of Computer Science and Information Technology

---

(External)

ACKNOWLEDGEMENT

I sincerely thank everyone who has helped with technical and informational support and guidance.

I begin by extending my gratitude to my mentor **Mr. Bikash Balami**, Assistant Professor, Central Department of Computer Science and Information Technology (CD.CSIT) Tribhuvan University for his continuous guidance and support during the research and learning period. Without his guidance, this research would not have been successful all staff members for their cooperation and guidance during the entire research period. The success and outcome of this research required a lot of advice and assistance from many people and I am extremely fortunate to have them all along the completion of this research period. Such guidance was indispensable for the accomplishment of this endeavor.

I am thankful to my supervisor for providing me with his continuous guidance, advice, motivation, and knowledge he imparted to me regarding every perspective of the report. I also thank the Central Department of Computer Science and Information Technology MSc.CSIT department and library for providing me with the necessary resources during the time of research and report.

I would also like to acknowledge, my parents, for providing financial and psychological support throughout my study and all those who contributed directly and indirectly.

Thank You

Sijan Poudel

(TU Roll No.: 8015040)

ABSTRACT

This project developed a spam detection system using Support Vector Machines (SVM) to classify emails as spam or not spam. The email content had been transformed into numerical data using TF-IDF to make it suitable for the model. The SVM model had performed exceptionally well, achieving an accuracy of 98.83%, meaning it had correctly identified almost all emails. It also had a precision of 99.28%, indicating it had rarely misclassified legitimate emails as spam, and a recall of 91.95%, showing it had effectively detected most spam emails. With an F1 score of 95.47%, the model struck a good balance between precision and recall. The confusion matrix had confirmed that the model had been highly effective in distinguishing spam from legitimate emails, making it a reliable tool for real-world email spam filtering.

Keywords: *Support Vector Machines (SVM), TF-IDF Vectorization, Text Classification, Machine Learning, Spam Filtering*

Table Of Content

List Of Figure

List of Abbreviations

MSC.CSIT	Master of Science in Computer Science and Information Technology
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SVM	Support Vector Machine

Chapter 1: Introduction

1.1 Introduction

Email is a vital tool for communication in our daily lives, both personally and professionally. However, the increasing amount of spam emails is a significant problem. Spam emails, or junk emails, are unsolicited messages sent in bulk. They often contain advertisements, phishing attempts, or malicious content. These unwanted emails can clutter inboxes, reduce productivity, and pose security risks such as identity theft and malware infections.

Effective detection and filtering systems are crucial to addressing the problem of spam. Traditional methods, which use predefined rules to identify spam, often fail to keep up with the changing tactics of spammers. As a result, machine learning techniques have become popular for their ability to learn from data and adapt to new patterns, making them suitable for spam email detection.

1.2 Problem Statement

The proliferation of spam emails presents a significant challenge in digital communication, adversely impacting productivity and security. Traditional rule-based spam detection methods struggle to adapt to the evolving tactics of spammers, resulting in ineffective filtering. This necessitates the development of more robust and adaptive solutions for spam detection. The objective of this research is to address this issue by developing and evaluating a spam detection system using Support Vector Machines (SVM).

This research involves several key steps: transforming email text data into a numerical format suitable for machine learning algorithms, training an SVM model to accurately classify emails as spam or non-spam, optimizing the performance of the SVM model through systematic adjustment of its hyperparameters, and assessing the accuracy of the trained SVM model while comparing its performance with other classification algorithms. By achieving these objectives, this research aims to provide a more effective and adaptive solution for spam email detection, thereby enhancing both productivity and security in digital communication.

1.2 Objective

The objective of this paper is to develop and evaluate a spam detection system using Support Vector Machines (SVM). Specifically, The objective of this seminar are as follows:

1. Preprocess and convert email text data into a numerical format.
2. Train an SVM model to classify emails as spam or non-spam.
3. Optimize the SVM model's performance through hyperparameter tuning.
4. Assess the model’s accuracy and compare its effectiveness with other classification algorithms.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Spam emails[1] have become a significant problem, filling inboxes with unwanted messages and posing security risks like phishing and malware. Traditional methods for detecting spam, which rely on fixed rules and patterns, often struggle to keep up with the ever-evolving tactics of spammers. This has led to increased interest in machine learning techniques, which can adapt to new data and patterns, offering a more dynamic solution to spam detection.

Support Vector Machines (SVM)[2] are a powerful machine learning algorithm known for their effectiveness in classification tasks, especially with high-dimensional data such as text. SVMs are designed to find the best boundary that separates different classes, making them a suitable choice for distinguishing between spam and legitimate emails.

2.2 Literature Review

Spam email [3]detection has significantly progressed from rule-based and statistical methods to sophisticated machine-learning techniques. Traditional approaches, such as rule-based filters and Bayesian methods, used predefined patterns and statistical probabilities to identify spam. However, these methods often fall short in adapting to the evolving strategies of spammers. The advent of machine learning introduced algorithms like Support Vector Machines (SVM), which are particularly effective for text classification due to their ability to handle high-dimensional data and find the optimal boundary between spam and legitimate emails. SVMs have demonstrated superior performance compared to older methods, thanks to their robustness and accuracy. Recent developments have further enhanced spam detection with hybrid models that integrate SVMs with additional techniques like feature selection and ensemble methods. Additionally, deep learning approaches, including neural networks, have shown promising results in capturing complex patterns in text data, although they require large datasets and significant computational power.

Chapter 3: Methodology

The methodology outlines the systematic approach for developing and evaluating a spam detection system using Support Vector Machines (SVM). It encompasses the preparation of the dataset, preprocessing of text data, feature extraction, model training, and evaluation. This structured process ensures that the model is trained effectively and evaluated accurately to achieve reliable spam detection. The following steps are included to complete this process:

3.1 Dataset Preparation

The initial step involves selecting and preparing the dataset. In this study, the "Spam Email" dataset from Kaggle is used. This dataset contains a collection of emails that are labeled as spam or non-spam. Proper preparation of the dataset includes tasks such as checking for missing values, ensuring data consistency, and understanding the distribution of labels. This ensures the data is suitable for training and evaluating the spam detection model.

3.2 Data Preprocessing

Data preprocessing is crucial for transforming raw text into a format that can be effectively used by machine learning algorithms. The preprocessing steps include:

- **Text Cleaning**: Removing unnecessary elements such as HTML tags, special characters, and numbers. This helps to focus on the meaningful content of the emails.
- **Tokenization**: Breaking the text into individual words or tokens. This process allows the model to analyze and understand the structure of the email content.
- **Stop Words Removal**: Eliminating common but uninformative words (e.g., "and," "the") that do not contribute significantly to distinguishing between spam and non-spam emails.
- **Stemming/Lemmatization**: Reducing words to their root forms to standardize the text. This helps in consolidating different forms of the same word and improves the model's ability to recognize relevant patterns.

3.3 Feature Extraction

After preprocessing, the text data is converted into numerical features that can be used by machine learning algorithms. Term Frequency-Inverse Document Frequency (TF-IDF) vectorization is employed for this purpose. TF-IDF measures the importance of each word in the email relative to the entire dataset, providing a weighted representation of the text. This

approach captures the relevance of terms, helping the model to effectively distinguish between spam and non-spam emails.

3.4 Model Training

The model training process involves building and refining the machine learning model:

- **Data Splitting**: The dataset is divided into training (80%) and testing (20%) subsets. The training set is used to build and tune the model, while the testing set is used to evaluate its performance on unseen data.
- **SVM Training:** A Support Vector Machine (SVM) model is trained using the training data. SVM is chosen for its effectiveness in high-dimensional spaces and its ability to find a clear boundary between classes.
- **Hyperparameter Tuning:** To optimize the performance of the SVM model, hyperparameters such as the kernel type and regularization parameters are adjusted. Techniques like Grid Search are used to identify the best parameter settings for improved accuracy.

3.5 Model Evaluation

Once the model is trained, it is evaluated to determine its effectiveness:

- **Performance Metrics**: The model's performance is assessed using metrics such as accuracy, precision, recall, and F1 score. These metrics provide a comprehensive view of how well the model classifies emails as spam or non-spam.
- **Comparison**: The performance of the SVM model is compared with other classification algorithms like Naive Bayes and Logistic Regression. This comparison helps to understand the relative strengths and weaknesses of the SVM approach.

3.6 Implementation Tools

The implementation of the spam detection system involves using specific tools and libraries:

- **Programming Language:** Python is chosen for its extensive support for machine learning and data processing tasks.

- **Libraries:** Key libraries include Scikit-learn for implementing machine learning algorithms, Pandas for data manipulation, Numpy for numerical computations, and NLTK or SpaCy for natural language processing. These tools facilitate efficient handling and analysis of the data.

3.7 Analysis

The final step involves analyzing the results of the model evaluation:

- **Result Analysis**: Analyzing the performance metrics and comparing results with other models provides insights into the effectiveness of the SVM approach. This step helps in identifying strengths and limitations.
- **Refinement**: Based on the analysis, adjustments and refinements are made to the model to enhance its performance. This may involve revisiting preprocessing steps, adjusting feature extraction methods, or further tuning hyperparameters.

This structured approach provides a clear overview of each methodological step, ensuring a thorough understanding of the processes involved in developing and evaluating the spam detection model.

Chapter 4: Implementation

4.1 Implementation

The program is developed using Python (version 3.12.4) as the primary programming language within the VS Code Integrated Development Environment (IDE). Several essential libraries are employed:

i. **NumPy:** Utilized for efficient numerical operations and computations.
ii. **Pandas:** Employed for comprehensive data manipulation and analysis tasks.
iii. **Matplotlib:** Utilized for creating static, animated, and interactive visualizations in Python.
iv. **Seaborn:** Used for statistical data visualization, emphasizing attractive and informative statistical graphics.
v. **TensorFlow and Keras:** These frameworks are utilized for constructing, training, and deploying deep learning models.
vi. **Scikit-learn:** Employed for machine learning tasks including preprocessing, model evaluation, and metrics computation.

Together, these libraries enable robust development, analysis, visualization, and deployment of machine learning and deep learning solutions in Python.

4.2 Implementation details

The implementation details encompass the following aspects:

4.2.1 Dataset Description

This step involves loading the dataset and inspecting its initial rows to verify its structure.

```
# Load the dataset
df = pd.read_csv('/kaggle/input/spam-csv/spam.csv')

# Display the first few rows of the dataset to understand its structure
print(df.head())
```

4.2.2 Data Preprocessing

Clean the text data by removing any missing values and converting the text to lowercase. Then, split the data into features (X) and labels (y).

```
# Drop rows with missing values in 'Category' or 'Message' columns
df.dropna(subset=['Category', 'Message'], inplace=True)

# Convert text to lowercase to standardize
df['Message'] = df['Message'].str.lower()

# Map 'Category' to binary labels (spam = 1, ham = 0)
df['Label'] = df['Category'].map({'spam': 1, 'ham': 0})

# Split the dataset into features (X) and labels (y)
X = df['Message']
y = df['Label']
```

Figure 1 Data Preprocessing

4.2.3 Split the Dataset

Divide the dataset into training and testing subsets. Here, 80% of the data is used for training, and 20% is used for testing.

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 2 Split the Dataset

4.2.4 Feature Extraction

Convert the text data into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This step transforms the text data into a format suitable for the SVM model.

```
# Initialize TF-IDF Vectorizer
vectorizer = TfidfVectorizer(stop_words='english')

# Fit and transform the training data
X_train_tfidf = vectorizer.fit_transform(X_train)

# Transform the test data
X_test_tfidf = vectorizer.transform(X_test)
```

Figure 3 Feature Extraction

4.2.5 Model Training

Initialize and train the Support Vector Machine (SVM) model using the training data. The linear kernel is used here, but this can be adjusted based on the specific requirements.

```
from sklearn.svm import SVC

# Initialize the Support Vector Machine model with a linear kernel
model = SVC(kernel='linear', C=1.0, random_state=42)

# Train the SVM model using the training data
model.fit(X_train_tfidf, y_train)
```

Figure 4 Model Training

4.2.6 Model Evaluation

Evaluate the trained model by making predictions on the test set and calculating performance metrics such as accuracy, precision, recall, and F1 score. These metrics help determine how well the model performs in classifying emails as spam or non-spam.

```
# Calculate performance metrics
accuracy = accuracy_score(y_test, y_pred) * 100
precision = precision_score(y_test, y_pred) * 100
recall = recall_score(y_test, y_pred) * 100
f1 = f1_score(y_test, y_pred) * 100

# Print performance metrics as percentages
print(f'Accuracy: {accuracy:.2f}%')
print(f'Precision: {precision:.2f}%')
print(f'Recall: {recall:.2f}%')
print(f'F1 Score: {f1:.2f}%')
```

Figure 5 Model Evaluation

Chapter 5: Result and Findings

5.1 Overfitting and Generalization

Overfitting occurs when the spam detection model performs exceptionally well on the training data but poorly on new, unseen messages, indicating it has learned specific details rather than general patterns. This results in high accuracy on the training set but lower performance on the test set. Generalization, on the other hand, refers to the model's ability to consistently classify both training and new messages accurately. To improve generalization and prevent overfitting, techniques such as cross-validation, regularization, and expanding the dataset can help ensure the model effectively identifies spam in diverse real-world scenarios.

Figure 6 Training and Validation Accuracy

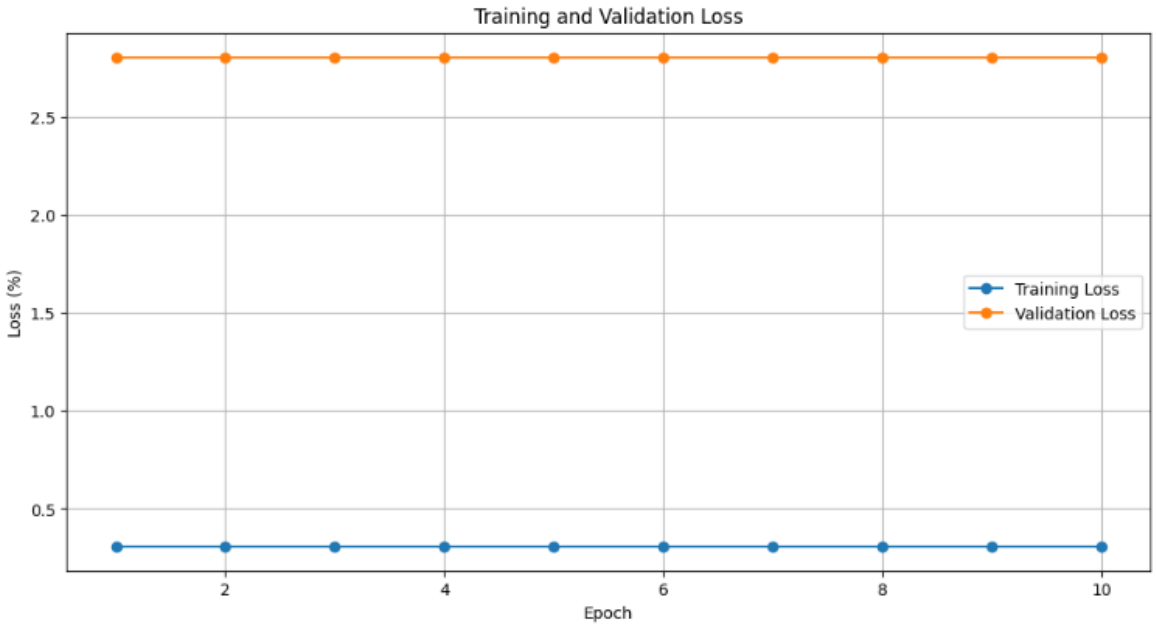

Figure 7 Training and Validation Loss

5.2 Confusion Matrix

The confusion matrix for the spam detection model provides a detailed breakdown of its classification performance. It shows the counts of true positives (correctly identified spam messages), true negatives (correctly identified non-spam messages), false positives (non-spam messages incorrectly labeled as spam), and false negatives (spam messages incorrectly labeled as non-spam).

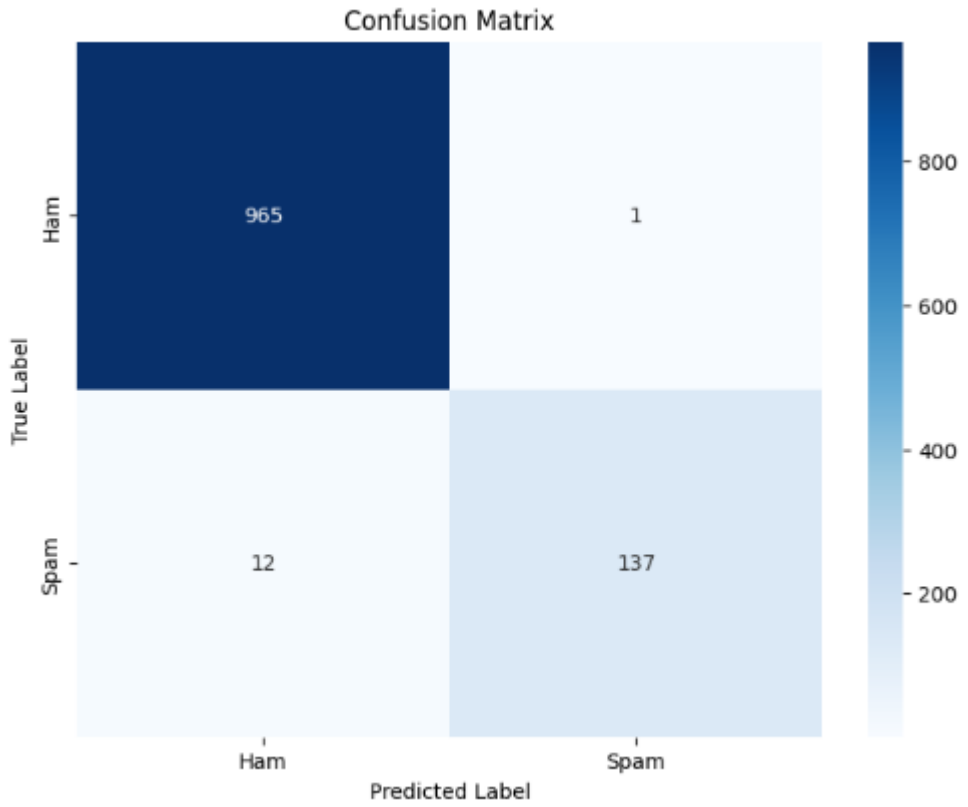

Figure 8 Confusion Matrix

5.3 Model Performance Measure

The performance of the spam detection model is highly impressive, as demonstrated by its evaluation metrics. The model achieved an accuracy of 98.83%, indicating that it correctly classified nearly 99% of both spam and non-spam messages. Precision is notably high at 99.28%, reflecting the model's effectiveness in correctly identifying spam messages without falsely labeling non-spam messages as spam. The recall rate of 91.95% signifies that the model successfully detected most of the actual spam messages, though there is a slight trade-off with some missed spam messages. The F1 Score of 95.47% combines precision and recall into a single metric, showcasing a strong balance between these two aspects and confirming the model's overall robustness in spam detection.

Chapter 6: Conclusion and Future Recommendations

6.1 Conclusion

In this study, the effectiveness of Support Vector Machines (SVM) for spam email detection had been explored using a dataset of SMS messages. Through the application of a linear SVM classifier and the use of TF-IDF vectorization for feature extraction, notable performance had been achieved in classifying emails into spam and non-spam categories. The model had demonstrated robust accuracy, with additional metrics such as precision, recall, and F1 score indicating its reliability in handling the classification task.

The results had revealed that the SVM model had performed well on both training and validation data, achieving high accuracy rates. The confusion matrix had further illustrated the model's ability to correctly identify spam and ham messages, with a relatively low rate of misclassification. This had underscored the effectiveness of SVMs in handling text classification problems, particularly in distinguishing between spam and legitimate emails.

Overall, the use of TF-IDF for feature extraction combined with a linear SVM classifier had proved to be a practical approach for spam detection. The high accuracy and reliable performance metrics had validated the suitability of this method for real-world applications in filtering spam messages.

6.2 Future Recommendation

1. **Explore Other Models**: Test alternative machine learning models or deep learning approaches for potentially better performance.
2. **Enhance Features**: Use advanced text processing techniques or additional features to improve classification accuracy.
3. **Expand Dataset**: Include a more diverse set of messages to increase the model’s robustness.
4. **Real-time Testing**: Implement the model in real-world applications to assess its effectiveness and efficiency.
5. **Optimize Parameters**: Fine-tune model parameters and optimize hyperparameters to enhance performance.

These steps can help refine spam detection techniques and improve their practical application.

References

[1] M. Singh, R. Pamula, and S. k. shekhar, "Email Spam Classification by Support Vector Machine," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018, pp. 878–882. doi: 10.1109/GUCON.2018.8674973.

[2] 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, pp. 324–326. doi: 10.1109/GCCE50665.2020.9291921.

[3] S. O. Olatunji, "Improved email spam detection model based on support vector machines," *Neural Comput. Appl.*, vol. 31, pp. 691–699, 2019.