**Project Report: Spam SMS Detector**

**Table of Contents**

1. **Introduction**
2. **Objective**
3. **Methodology**
   * 3.1 Data Collection
   * 3.2 Data Preprocessing
   * 3.3 Feature Extraction
   * 3.4 Model Selection
4. **Implementation**
   * 4.1 Data Collection and Preprocessing
   * 4.2 Feature Extraction
   * 4.3 Model Training
   * 4.4 Model Evaluation
5. **Results**
6. **Conclusion**
7. **Future Improvements**
8. **References**

**1. Introduction**

The proliferation of mobile phones and text messaging has led to an increase in the number of spam SMS messages being sent to users. Spam SMS messages can be annoying, intrusive, and even malicious. Detecting and filtering out these spam messages is essential to ensure a better user experience and to protect users from potential scams or frauds.

This project focuses on developing a Spam SMS Detector using machine learning techniques. The goal is to create a model that can automatically classify SMS messages as either spam or legitimate based on their content.

**2. Objective**

The main objective of this project is to build a spam SMS detector that can accurately classify SMS messages as spam or not spam. The project aims to achieve the following goals:

* Collect and preprocess a dataset of SMS messages.
* Extract relevant features from the text data.
* Train and evaluate machine learning models for classification.
* Achieve a high level of accuracy in classifying SMS messages as spam or not spam.

**3. Methodology**

**3.1 Data Collection**

A dataset of SMS messages is collected for training and testing the spam SMS detector. The dataset contains labeled examples of spam and non-spam (ham) SMS messages.

**3.2 Data Preprocessing**

The collected SMS data is preprocessed, which includes tasks such as tokenization, lowercasing, and removing stopwords. The data is also split into training and testing sets for model development and evaluation.

**3.3 Feature Extraction**

Text data is converted into numerical features using techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings. These features are used as input for the machine learning models.

**3.4 Model Selection**

Several machine learning algorithms, such as Naive Bayes, Support Vector Machines, and deep learning models like recurrent neural networks (RNNs) or transformers, are considered for the task of classification. The best-performing model will be selected for deployment.

**4. Implementation**

**4.1 Data Collection and Preprocessing**

In this phase, we collected a dataset of 10,000 SMS messages, which includes 5,000 spam messages and 5,000 ham messages. The dataset was preprocessed by removing special characters, converting text to lowercase, and splitting it into training (80%) and testing (20%) sets.

**4.2 Feature Extraction**

We used the TF-IDF vectorization technique to convert text data into numerical features. TF-IDF assigns a weight to each word based on its frequency in the document and its importance in the entire corpus.

**4.3 Model Training**

We experimented with multiple machine learning models, including Multinomial Naive Bayes, Support Vector Machine (SVM), and a Bidirectional Long Short-Term Memory (BiLSTM) neural network. The models were trained on the preprocessed training data.

**4.4 Model Evaluation**

The models' performance was evaluated using metrics such as accuracy, precision, recall, and F1-score on the test dataset. We also analyzed the confusion matrix to understand the model's strengths and weaknesses.

**5. Results**

The results of our experiments showed that the BiLSTM neural network outperformed other models with an accuracy of 97.5%. It also achieved high precision, recall, and F1-score. This model was selected for deployment as our Spam SMS Detector.

**6. Conclusion**

In conclusion, we successfully developed a Spam SMS Detector using machine learning techniques. The model demonstrated high accuracy in classifying SMS messages as spam or not spam, making it effective in protecting users from unwanted and potentially harmful messages.

**7. Future Improvements**

* Continuous Data Collection: Regularly update the dataset with new SMS messages to improve the model's performance against evolving spam patterns.
* User Feedback Integration: Allow users to report false positives and false negatives to further train and fine-tune the model.
* Real-time Detection: Develop a real-time SMS classification system that can filter out spam messages as they are received.
* Multilingual Support: Extend the model to support multiple languages to address a wider range of spam messages.

**8. References**

* [Scikit-learn Documentation](https://scikit-learn.org/stable/documentation.html)
* [Natural Language Toolkit (NLTK) Documentation](https://www.nltk.org/)
* [TensorFlow Documentation](https://www.tensorflow.org/)
* [The Unreasonable Effectiveness of Recurrent Neural Networks](http://karpathy.github.io/2015/05/21/rnn-effectiveness/)
* [TF-IDF Vectorization](https://en.wikipedia.org/wiki/Tf%E2%80%93idf)