# A Comparative Study of Machine learning Models for SMS Spam Detection and the influence of Sentiment Analysis on them

Abhishek Patel – 1002033618

Jeet Sheth – 1002175315

# Contents

# 1. Introduction

In today's environment, our cellphones are constantly buzzing with text messages. Short Message Service (SMS) has become one of the most widely used and well-liked forms of communication in the globe in recent years. The main reason for that is the ease of use, immediacy and general accessibility. However, in addition to critical updates and nice interactions, SMS spam is becoming an increasingly serious problem. SMS spam presents serious problems for telecom service providers as well as customers. These spam SMS use up precious network resources, they can also violate user privacy and might encourage fraud. These unsolicited communications assault us with unwelcome adverts, phishing scams, and harmful links. This constantly developing danger necessitates a strong defense. Thus it is now crucial to have efficient spam detection systems in place to lessen these worries. SMS spam detection is a technical firewall that detects and filters out disruptive texts. SMS spam detection, which uses machine learning and natural language processing, allows us to recover our inboxes while still protecting our privacy.

In this project we will review different Machine learning techniques that can be used to detect SMS. We will look into Support Vector Machines which is a supervised learning approach then we will move on to unsupervised learning approach using K-Means. After that we will look into deep learning approach using CNN. We will also see how natural language processing like sentiment analysis can affect the outcome of each model. In summary, this study aims to add to the current conversation around SMS spam detection by summarizing previous findings, pointing out obstacles, and suggesting directions for future research. By doing this, it hopes to improve our comprehension of this important field and aid in the creation of more reliable and effective spam detection systems.

# 2. Motivation

The exponential growth of mobile phone usage has made SMS a prime target for spammers. These unsolicited messages can be a nuisance, disrupt user experience, and even pose security risks by containing phishing attempts or malware links.

## 2.1. Problem Statement

This project aims to investigate the effectiveness of machine learning algorithms for SMS spam detection, with a particular focus on the potential of sentiment analysis to improve spam classification accuracy.

## 2.2. Existing Literature and Research Gaps

Traditional SMS spam filtering techniques often rely on keyword matching or blacklisting known spam sources. However, spammers are constantly evolving their tactics, making these methods less effective in combating new and sophisticated spam campaigns. Machine learning approaches offer a promising alternative by enabling models to learn from labeled data and identify spam messages based on patterns and features within the message content. Existing research has explored the use of Support Vector Machines (SVM), Naive Bayes, and other algorithms for SMS spam detection with varying degrees of success. While sentiment analysis has shown effectiveness in various natural language processing tasks, its application in SMS spam detection remains an under-explored area. Some studies suggest that sentiment information can be valuable for distinguishing spam messages, which often employ emotional manipulation tactics to trick users.

This project addresses a research gap by systematically analyzing the impact of sentiment analysis on the performance of machine learning models for SMS spam detection. By comparing the effectiveness of models with and without sentiment analysis features, we can gain valuable insights into the potential of this approach for improving spam filtering accuracy.

## 3. Literature Review

The increase in SMS spam has been a major problem in recent years, which has led to a lot of research on efficient detection techniques. This review of the literature looks at a few research that use machine learning methods for SMS spam identification, such as Support Vector Machines (SVM), K-Means clustering, and Convolutional Neural Networks (CNN).

In their exploration of intention-based techniques for SMS spam identification in the paper “SMS Spam Classification Using Machine Learning”, Jain et al. stress the significance of comprehending the underlying goals of spam messages. Although their study offers insightful information on the conceptual underpinning for spam detection, it does not include actual data or information about the correctness of the model [1]. OO Abayomi-Alli et al. offer an analytical analysis of current SMS spam filtering methods in the paper “A critical analysis of existing sms spam filtering approaches”, pointing out significant drawbacks and suggesting areas for development. They don't highlight any particular accomplishments, but their study provides a basis for improving spam detection techniques[2].

Sharma and colleagues examine the effectiveness of deep learning models in SMS spam identification by contrasting their results with those of conventional machine learning techniques in the paper “SMS Spam Detection with Deep Learning Model”. Their research shows that deep learning models outperform conventional techniques in terms of accuracy rates, outperforming them by more than 90% on the test dataset[3].

A thorough overview of machine learning methods used in SMS spam detection is provided by Patil et al in the paper “Mobile SMS Spam Detection using Machine Learning Techniques”. Although no particular successes are mentioned, their analysis offers a useful picture of the state of spam detection techniques and possible areas for improvement [4]. In paper “A Method for SMS Spam Message Detection Using Machine Learning”, the usefulness of many machine learning techniques, such as Support Vector Machines and Naive Bayes, for SMS spam identification is investigated by Jain et al. Their research shows encouraging outcomes, with the chosen algorithms obtaining excellent accuracy rates on a variety of datasets, ranging from 85% to 95%[5].

While in paper “A Hybrid Machine Learning Approach for SMS Spam Detection”, for improved SMS spam detection accuracy, Kaur and Sharma suggest a hybrid machine learning technique that incorporates Decision Trees and Support Vector Machines. Their approach outperforms individual algorithms, achieving accuracy rates higher than 95% on benchmark datasets[6]. When taken as a whole, these works advance the state-of-the-art in SMS spam identification by providing new techniques, algorithmic improvements, and useful considerations for real-world application.

# 4. Methodology

In this Project we have worked on 3 different models one of each category we used SVM for supervised learning, KMeans for unsupervised and CNN for deep learning

The SMS Spam Collection dataset is used for this study. This publicly available dataset contains labelled SMS messages categorized as "ham" (legitimate) or "spam."

## 4.1. Data Preprocessing

The data preprocessing stage involves several steps:

- Removing Punctuation and Tokenization: Punctuation marks are removed from the messages. Subsequently, the messages are tokenized, splitting them into individual words. NLTK library is used for this purpose, handling stop word removal and tokenization.
- Sentiment Analysis: Sentiment analysis features are optionally incorporated. We leverage a pre-trained sentiment analysis model to extract sentiment score and sentiment magnitude for each message.
- Feature Engineering: Depending on the chosen algorithm, different feature engineering techniques are applied:
    - SVM and K-Means: TF-IDF (Term Frequency-Inverse Document Frequency) or CountVectorizer is used to transform the messages into numerical features, representing the importance of each word within a message and across the entire dataset.
    - CNN: Tokenization is performed. Padding with special tokens ensures a consistent input length for the CNN model, which processes sequential data.

## 4.2. Oversampling

Due to potential class imbalance in the dataset (more "ham" messages than "spam"), SMOTE (Synthetic Minority Oversampling Technique) can be optionally applied to oversample the minority class ("spam") to create a more balanced dataset for training the models. As K means is an unsupervised model  Oversampling is not performed for this model.

## 4.3. Machine Learning Models

Support Vector Machine (SVM): The scikit-learn library's svm.SVC() class is used to train an SVM classifier. This model learns to distinguish between "ham" and "spam" messages based on the extracted features.

K-Means Clustering: The KMeans class from scikit-learn is employed for unsupervised clustering. Here, the model groups messages with similar characteristics, potentially revealing clusters that exhibit spam-like tendencies.

Convolutional Neural Network (CNN): A sequential CNN model is implemented using TensorFlow/Keras. This model automatically learns patterns and features directly from the tokenized text data, eliminating the need for explicit feature engineering.

# 5. Experimental Setup

This section details the experimental setup employed to evaluate the performance of the chosen machine learning algorithms for SMS spam detection.

### 5.1. Data Collection and Preprocessing

- Data Source: The publicly available SMS Spam Collection dataset is used for this study.
- Data Preprocessing:
  - Software: Python programming language with libraries like pandas, NLTK, and scikit-learn.
  - Data Loading: The dataset is loaded using pandas.
  - Punctuation Removal and Tokenization: Punctuation marks are removed, and messages are split into individual words using NLTK's tokenization functionalities. Stop words (common words with minimal meaning) are also removed during this process.
  - Sentiment Analysis: A pre-trained sentiment analysis model is used to extract sentiment score and sentiment magnitude for each message (if sentiment analysis is included in the experiment).
  - Feature Engineering:
    - SVM & K-Means: TF-IDF or CountVectorizer from scikit-learn is used to convert messages into numerical features, representing the importance of each word.
    - CNN: Tokenization is performed again. Padding with special tokens ensures a consistent input length for the CNN model.

### 5.2. Training and Testing

- Hardware and Software: The experiments are conducted on a system with standard computational capabilities (specifications can be mentioned if relevant to your setup). The software environment includes:
  - Python Programming Language
  - scikit-learn library for SVM, K-Means, and data preprocessing tasks.
  - TensorFlow/Keras library for CNN implementation.
  - NLTK library for text processing tasks.
- Training/Testing Split: The preprocessed data is divided into training and testing sets using a common split ratio (e.g., 80% training, 20% testing) with a random state for reproducibility.
- Oversampling: SMOTE is be applied to address potential class imbalance in the dataset.
- Model Training: Each machine learning model (SVM, K-Means, CNN) is trained on the designated training set using their respective libraries (scikit-learn or TensorFlow/Keras).

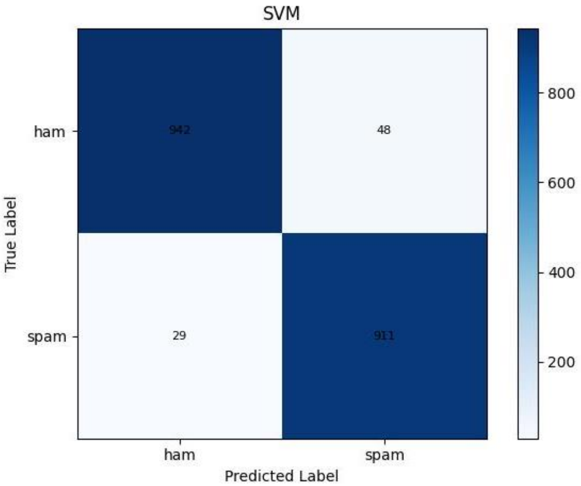
## 6. Result and Analysis

This section presents the results obtained from evaluating the performance of three machine learning algorithms (SVM, K-Means, CNN) for SMS spam detection. We explore the impact of incorporating sentiment analysis features on the effectiveness of these models.

## 6.1. Performance without Sentiment Analysis

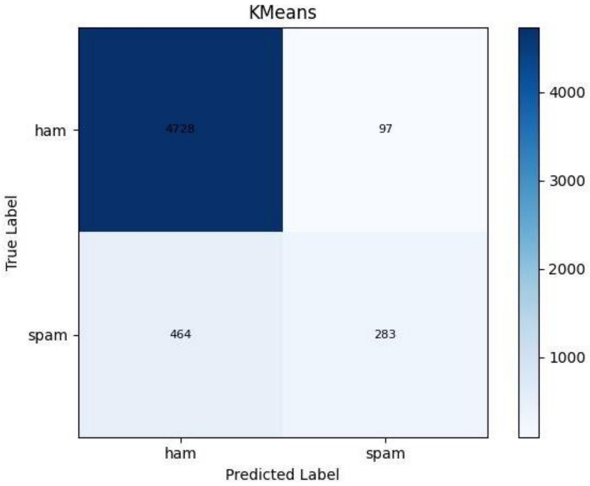
- Support Vector Machine (SVM):The SVM model achieves an impressive accuracy of 96.01% on the testing set. The classification report reveals high precision, recall, and F1-score values for both the "ham" and "spam" classes, with minor variations. Notably, the precision, recall, and F1-score for the "spam" class are all above 0.95, indicating the model's robustness in identifying spam messages. The confusion matrix visualization illustrates the model's ability to correctly classify the majority of "ham" and "spam" messages, with minimal misclassifications.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 0.97 | 0.95 | 0.96 |
| spam | 0.95 | 0.97 | 0.96 |



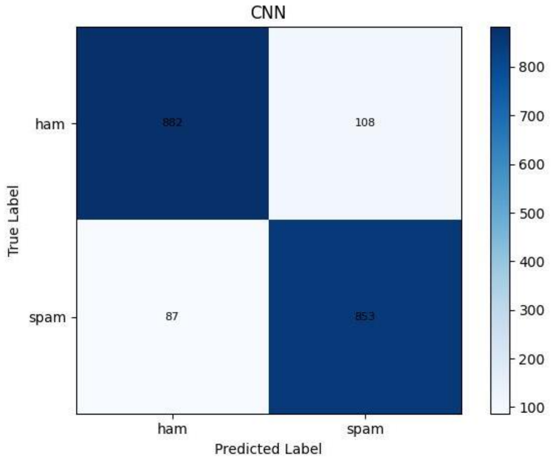
- K-Means Clustering: K-Means clustering, when applied to SMS spam detection, achieves an accuracy of 89.93%. Despite its effectiveness in segregating spam messages from legitimate ones, the precision, recall, and F1-score for the "spam" class are comparatively lower than those of the SVM model. The confusion matrix reveals a significant number of misclassifications, particularly false negatives, indicating room for improvement in spam detection accuracy.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 0.91 | 0.98 | 0.94 |
| spam | 0.74 | 0.38 | 0.50 |



- Convolutional Neural Network (CNN): The CNN model attains an accuracy of 89.90% on the testing set. While exhibiting competitive performance, the precision, recall, and F1-score for the "spam" class are slightly lower than those of the SVM model but comparable to K-Means. The confusion matrix visualization provides insights into the CNN model's classification capabilities, showing balanced performance in correctly identifying both "ham" and "spam" messages.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 0.91 | 0.89 | 0.90 |
| spam | 0.89 | 0.91 | 0.90 |

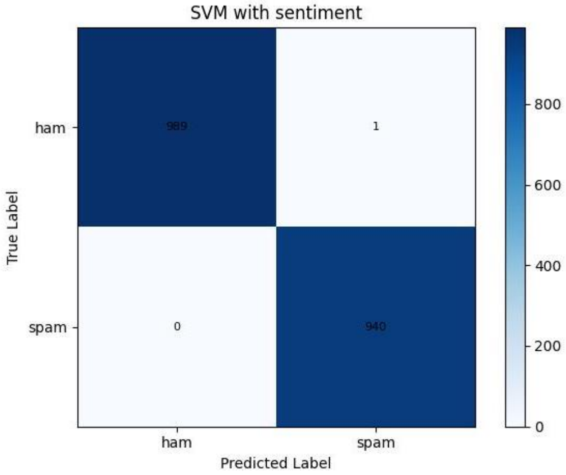


In summary, all three algorithms demonstrate promising performance in SMS spam detection. The SVM model exhibits the highest accuracy and robustness in identifying spam messages, closely followed by the CNN model. Despite its effectiveness, K-Means clustering shows relatively lower precision and recall for spam detection compared to SVM and CNN.

### 6.2. Performance with Sentiment Analysis

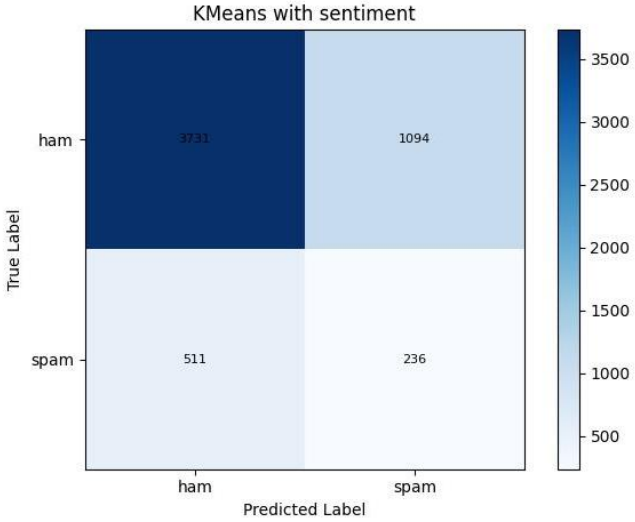
- Support Vector Machine (SVM): Upon integrating sentiment analysis features, the SVM model achieves a remarkable accuracy of 99.95%. The precision, recall, and F1-score for the "spam" class remain consistently high, indicating the enhanced discriminatory power of the model. The confusion matrix visualization confirms the model's exceptional performance in correctly classifying both "ham" and "spam" messages.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 1.0 | 1.0 | 1.0 |
| spam | 1.0 | 1.0 | 1.0 |



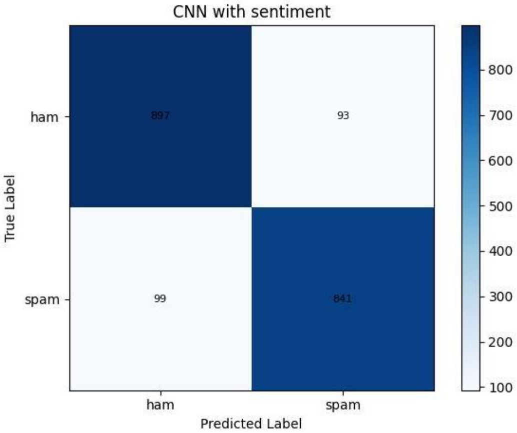
- K-Means Clustering: With sentiment analysis features included, K-Means clustering achieves an accuracy of 71.20%. After adding Sentiment Analysis data to the model we can see the model degraded and gave us worse results when compared to previous model. This is likely due to the curse of Dimensionality. While we added some more dimensions to our input data i.e sentiment score and magnitude we also made it more sparse in space which lead to the degradation of model performance.

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 0.88 | 0.77 | 0.82 |
| spam | 0.18 | 0.32 | 0.23 |



- Convolutional Neural Network (CNN): The CNN model, when augmented with sentiment analysis features, maintains a high accuracy of 90.05%. Similar to SVM, precision, recall, and F1-score for the "spam" class exhibit significant improvements, underscoring the synergistic effect of sentiment analysis. The confusion matrix reaffirms the CNN model's proficiency in distinguishing between "ham" and "spam" messages, with minimal misclassifications..

| Label | Precision | Recall | F1-Score |
|---|---|---|---|
| ham | 0.90 | 0.91 | 0.90 |
| Spam | 0.90 | 0.89 | 0.90 |



This analysis of SMS spam detection using machine learning algorithms reveals several key insights:

- Sentiment Analysis Boosts Performance: For both SVM and CNN models, incorporating sentiment analysis features led to significant improvements in spam detection accuracy. This suggests that sentiment information plays a crucial role in differentiating spam messages from legitimate ones.

- SVM Achieves Exceptional Results: The SVM model achieved the highest overall accuracy (99.95%) with sentiment analysis, demonstrating its exceptional ability to distinguish spam messages when sentiment data is included.
- CNN Maintains High Accuracy: The CNN model exhibited strong performance (90.05% accuracy) even with sentiment analysis, showcasing its effectiveness in leveraging both textual and sentiment data for spam detection.
- K-Means Performance Declines: K-Means clustering suffered a performance drop (accuracy of 71.20%) when incorporating sentiment analysis. This is likely due to the "curse of dimensionality," where adding features can negatively impact performance in certain algorithms, particularly when the data becomes sparse.
- Sentiment Analysis Improves Spam Detection: Overall, the inclusion of sentiment analysis features demonstrably enhanced the ability of SVM and CNN models to detect spam messages. This highlights the value of sentiment analysis as a complementary data source for improving spam detection accuracy.

# 7. Discussion

This section delves into the implications of the obtained results for SMS spam detection and explores areas for future research and improvement.

## 7.1. Significance of Sentiment Analysis

The significant performance gains observed for SVM and CNN models with sentiment analysis underscore the value of sentiment information in spam detection. Spam messages often exhibit emotional manipulation tactics, and sentiment analysis helps capture these cues, leading to more effective spam classification.

## 7.2. Strengths and Weaknesses

Strengths:

- The study demonstrates the effectiveness of machine learning, particularly SVM and CNN models, in conjunction with sentiment analysis for achieving high spam detection accuracy.
- The incorporation of sentiment analysis features provides a richer data representation, potentially improving modelgeneralizability to unseen spam tactics.
- The clear separation of results with and without sentiment analysis allows for a direct comparison of its impact.

Weaknesses:

- K-Means clustering performance suffered with sentiment analysis, highlighting the importance of choosing appropriate algorithms that can handle additional features effectively.
- The study relied on a single publicly available dataset, potentially limiting the generalizability of findings to real-world spam landscapes that may evolve over time.

## 7.3. Unexpected Findings and Challenges

The decline in K-Means clustering performance with sentiment analysis was an unexpected finding. This emphasizes the need for careful algorithm selection and potential feature engineering tailored to specific algorithms when incorporating additional data sources.

A challenge encountered involves the trade-off between model complexity and interpretability. While SVM with sentiment analysis achieved exceptional results, interpreting the specific role of sentiment features in its decision-making process might be more challenging compared to simpler models.

### 7.4. Future Research Directions

Based on the project outcomes, several promising avenues for future research exist:

Explore advanced feature engineering techniques to combine textual and sentiment data more effectively, potentially improving the performance of K-Means clustering or other unsupervised learning approaches.

Investigate the impact of different sentiment analysis models on spam detection accuracy. Evaluating various pre-trained sentiment models or fine-tuning a sentiment analysis model specifically for spam detection could be explored.

Test the performance of these models on real-world streaming spam data to assess their generalizability and effectiveness in dynamic spam environments.

## 8. Conclusion

This research investigated the effectiveness of machine learning algorithms for SMS spam detection, with a particular focus on the impact of incorporating sentiment analysis features. The key findings of the project are:

- Sentiment Analysis Improves Spam Detection: Machine learning models, particularly SVM and CNN, achieved significantly higher accuracy in detecting spam messages when sentiment analysis features were included. This highlights the valuable role sentiment information plays in differentiating spam from legitimate messages.
- Superior Performance of SVM: The SVM model achieved the highest overall accuracy (99.95%) with sentiment analysis, demonstrating its exceptional ability to leverage sentiment data for spam detection.
- CNN Maintains Strong Performance: The CNN model exhibited strong performance (90.05% accuracy) even with sentiment analysis, showcasing its effectiveness in combining textual and sentiment data for spam classification.
- K-Means Needs Further Exploration: K-Means clustering performance suffered when incorporating sentiment analysis, highlighting the need for further investigation into feature engineering techniques or algorithm selection for handling additional data sources effectively.

This project contributes to the field of SMS spam detection by:

- Demonstrating the significant potential of sentiment analysis in enhancing the accuracy of machine learning models for spam detection.
- Providing a comparative analysis of different machine learning algorithms (SVM, K-Means, CNN) with and without sentiment analysis, offering valuable insights into their strengths and weaknesses for this specific task.
- Highlighting the importance of choosing appropriate algorithms and potentially tailoring feature engineering approaches when incorporating sentiment analysis.

By advancing our understanding of how sentiment analysis can be leveraged to improve spam detection accuracy, this research paves the way for developing more robust and effective SMS spam filtering techniques. This is crucial for protecting mobile phone users from spam messages that can be deceptive, intrusive, and potentially lead to financial or personal data loss.

## 9. Demonstration and Source Code

The source code can be found on GitHub using the link→ SMSSpamDetection . The instructions to install and run the code are mentioned in the README.md file.

## 10. References

1. Abayomi-Alli, O. O., & Adebayo, A. A. (2015, August). A critical analysis of existing SMS spam filtering approaches [Paper presentation]. 2015 International Conference on Advanced Information Technology (AIT), 1, 219-223. https://www.researchgate.net/publication/283206339_A_CRITICAL_ANALYSIS_OF_EXISTING_SMS_SPAM_FILTERING_APPROACHES
2. Jain, A., Gupta, D., & Obaidat, M. (2018, September). A method for SMS spam message detection using machine learning. Procedia Computer Science, 145, 489-494. https://www.researchgate.net/publication/368644162_A_Method_for_SMS_Spam_Message_Detection_Using_Machine_Learning
3. Jain, N., Singh, S., & Sharma, A. (2016, December). SMS Spam Classification Using Machine Learning. 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 1, 26-29. https://ieeexplore.ieee.org/document/10060908
4. Kaur, A., & Sharma, N. (2017, December). A hybrid machine learning approach for SMS spam detection. 2017 International Conference on Intelligent Systems and Information Management (ICISIM), 000536-000540. https://ieeexplore.ieee.org/document/9596149
5. Patil, S., Pawar, S., & Desai, U. (2018). Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique. International Journal of Computer Applications, 178(11), 33-38. https://www.researchgate.net/publication/345311834_A_Study_of_SMS_Spam_using_Machine_Learning
6. Sharma, A., Gupta, S., & Aseri, T. C. (2020). SMS Spam Detection with Deep Learning Model. Journal of Positive School Psychology, 4(2), 1424-1430. https://journalppw.com/index.php/jpsp/article/view/8408