Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

B.Tech. – Computer Science and Engineering (C.S.E.)

Course Code : BCSE354E

Course Title : Information Security Management

Course Faculty : Padmanaban R.

Slot : TB1

SMS Spam Filtering using Machine Learning and Deep Learning

Submitted By :

- Krish Goel – 21BCE6097
- Meghna Das – 21BCE1422

SMS SPAM FILTERING USING MACHINE LEARNING AND DEEP LEARNING

Krish Goel
Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT University)
Chennai, India
krish.goel2021@vitstudent.ac.in

Meghna Das
Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT University)
Chennai, India
meghna.das2021@vitstudent.ac.in

***Abstract*—Due to the ubiquity of mobile phones and the extensive usage of Short Message Service (SMS), SMS Spam has become a major annoyance to smartphone users [1]. In this research, we propose a complete solution that combines state-of-the-art Deep Learning techniques with standard Machine Learning approaches to filter SMS spam. To extract pertinent features from SMS texts, such as lexical, syntactic, and semantic properties, we first investigate feature engineering and selection strategies [2]. Then, using these features, traditional Machine Learning models like Random Forests, Naïve Bayes, and Support Vector Machines (SVMs) are trained [3]. Our suggested approach offers a scalable and effective way to filter SMS spam, giving customers better defense against unsolicited messages while minimizing false positives and maximizing classification accuracy. This work advances the field of spam detection systems and has important ramifications for improving security and user experience in mobile networks.**

Keywords—ubiquity, SMS (Short Message Service), spam, smartphone, state-of-the-art, deep learning, machine learning, unsolicited, false positives, ramifications.

I. INTRODUCTION

A rising number of people now consider their mobile phones to be their constant companions [4]. The commercial SMS (Short Message Service) market is now worth millions of dollars. As of early 2013, its value ranged from 11.3 to 24.7 percent of the Gross National Income (GNI) of developing nations [5]. Short Messaging Service (SMS) providers enable users of mobile phones to send and receive brief text messages. Unsolicited commercial adverts rose in volume as the site gained prominence [6]. The proliferation of mobile devices used for e-mail and messaging has led to an upsurge in spam. Currently, mobile consumers receive 85% of their e-mails and texts as spam. The cost of sending e-mailsand communications is quite low, but the cost of receiving them is significant. Spam and service provider costs can be quantified by the amount of time lost and essential messagesor emails that are lost [7]. Because each user has limited time, memory, and internet access, the value of emails and messages.

E-mail and text message Spam Filtering are very different inseveral key aspects. Unlike e-mails, which have many large datasets available, there are very few real databases for SMS spam [8]. Furthermore, because text messages are shorter than e-mails, there is far less information in them that can be used for classification. There isn't a header here either. Furthermore, text messages are filled with acronyms and significantly less formal language than e-mails. All these features have the potential to severely reduce the efficacy of widely used e-mail Spam Filtering algorithms when applied to short text messages [9].

Predicting SMS spam has long been a prominent area of research. The goal is to apply various Machine Learning algorithms to the problem of SMS spam classification, assess each one's efficacy to gain insight and further explore the matter, and develop an application built on an algorithm that can reliably filter SMS spam [10]. In the current work, several Machine Learning and Deep Learning-based predictive modelsare proposed to accurately estimate the movement of SMS spam. To further increase the models' capacity for prediction, the predictive framework incorporates a potent Deep Learning-based Long Short-Term Memory (LSTM) network [11].

II. LITERATURE REVIEW

1. Spam Detection Approach for Secure Mobile Message Communication, Shah Nazir, Habib Ullah Khan, and Amin-Ul-Haq, 2020

- They proposed the applications of the Machine Learning-based spam detection method for accurate detection. They also used Logistic Regression, K-Nearest Neighbor, and Decision Tree for the classification of ham and spam messages. Their LR model had an accuracy of 99 %.

2. Spam Filtering in SMS using Recurrent Neural Networks, Pumrapee Poomka, Wappana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop, 2019

- They used Natural Language Processing (NLP) and Deep Learning (DL) techniques. The model built using LSTM has an accuracy of 98.18 %.

3. SMS Spam Filtering : A Hybrid Approach using Machine Learning and Lexical Analysis, R. Vijayalakshmi, and Dr. T. Meyyappan, 2019

- In order to filter SMS spam, this research suggests a hybrid method that combines lexical analysis and Machine Learning techniques. The research leverages the advantages of both methodologies to improve spam detection's precision and resilience. The efficacy of the hybrid strategy is demonstrated by experimental findings on a large SMS dataset, which demonstrate considerable gains in classification performance compared to individual techniques.

4. An Effective SMS Spam Filtering Technique using Machine Learning Approaches, S. Saranya, R. Saranya, and Dr. V. Saravanan, 2018

- This study presents a Machine Learning-based method for efficiently filtering spam SMS messages. The study evaluates the accuracy of spam message identification using Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest

Neighbor (KNN) classifiers. The superiority of SVMs in terms of accuracy and efficiency is demonstrated by experimental results.

5. SMS Spam Detection using Deep Learning Techniques, S. Tiwari, and A. Khare, 2018

- The use of Deep Learning algorithms for SMS spam detection is investigated in this research. The study contrasts the performance of the Deep Learning architecture with that of conventional Machine Learning methods, based on Bi-directional Long Short-Term Memory (Bi-LSTM) networks. The outcomes of the experiments show that the Bi-LSTM model is more effective than other models at correctly classifying spam messages, suggesting that it could be used in practical SMS Spam Filtering scenarios.

6. SMS Spam Detection using Machine Learning Techniques : A Case Study, A. Joshi, and V. Sharma, 2018

- This research uses Machine Learning approaches to offer a case study of SMS spam identification. Using a sizable SMS dataset, the study evaluates the effectiveness of classifiers including Random Forest, Decision Trees, and Logistic Regression. The outcomes of the experiments show that Random Forest can achieve excellent recall and accuracy rates, which makes it suitable for real-world SMS Spam Filtering applications.

7. SMS Spam Detection Using Machine Learning Techniques : A Comparative Study, R. Gupta, S. Shukla, and A. K. Gupta, 2017

- A comparative analysis of Machine Learning methods for SMS spam detection is presented in this research. Using a sizable SMS dataset, the study assesses the effectiveness of classifiers including SVMs, Random Forest, and Gradient Boosting Machines (GBM). Based on experimental results, SVMs is a viable method for real-world SMS Spam Filtering applications since it can achieve high accuracy and F1-score.

8. A Novel Approach for SMS Spam Filtering using Machine Learning Techniques, Bhavana N. Mehta, and Vijay M. Gulhane, 2016

- This research provides a novel Machine Learning-based method for SMS Spam Filtering. The study applies several classification algorithms, such as Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR), and makes use of feature selection techniques to find pertinent features from SMS texts. The outcomes of the experiments demonstrate how well the suggested method works to differentiate between messages that are spam and those that are not.

9. SMS Spam Filtering : A Review of Machine Learning Techniques, Y. Osareh, and E. Khreich, 2016

- This research offers a thorough analysis of Machine Learning-based SMS Spam Filtering methods. The study looks at several feature extraction techniques, classification algorithms, and assessment criteria used in previous studies. In order to help academics and practitioners create spam detection systems that are more effective, it also addresses the difficulties and potential paths in SMS Spam Filtering research.

10. An Ensemble Learning Approach for SMS Spam Detection, S. Mehmood, U. Qamar, and S. Khan, 2015

- In order to enhance classification performance, this research suggests an ensemble learning method for SMS spam detection that combines many base classifiers. The work builds different classifier ensembles using methods like boosting and bagging, then tests their efficacy on an actual SMS dataset. The ensemble approach outperforms individual classifiers in accurately recognising spam messages while minimising false positives, as demonstrated by the experimental findings.

III. Proposed Methods

A. Classical Machine Learning Classifiers

In this section, we briefly discuss the six used Machine Learning classifiers : Naïve Bayes, Generalized Linear Model (GLM), Fast Large Margin, Decision Tree, Random Forest, Gradient Boosted Trees and Support Vector Machines [12]. A Naïve Bayes classifier is a framework used for probabilistic Machine Learning classification tasks. It is easy to use but computationally cost effective. Naïve Bayes' fundamental assumption is that, given the tag (class) value, the value of any attribute is independent of the value of any other attribute. GLM estimates models of regression for results after exponential distributions [13]. These include the Poisson, Binomial, and Gamma distributions in addition to the Gaussian distribution. Each serves a different purpose and can be used either for prediction or classification depending on the choice of distribution and connection function [27]. GLM is considered a dynamic version of linear regression models. Fast Large Margin is an SVM-like algorithm which runs in O(N) [14]. Because of its complexity, Fast Large Margin is perfect for classifying big data. A Decision Tree is a flowchart-like architecture; it can be used to represent decisions and decision-making visually and clearly [23]. Each part of a Decision Tree has a role in the classification process; the inner nodes represent checking of attributes, edges represent the result of the checking and the terminal nodes represent class labels [16]. Random Forest model is developed from Decision Trees. The primary idea of a Random Forest is merging a number of Decision-making Trees into one model. Separately, the findings of Decision Trees may lead to non-perfect results, but in combination, the findings are enhanced. Gradient Boosted Trees have the same idea as that of Random Forest models; but the difference is that in Gradient Boosted models, the combination task starts at the beginning [26]. If the parameters are carefully adjusted, Gradient models may produce better results than Random Forests. The disadvantage of Gradient models is that they suffer from noisy data. SVM is a popular and simple supervised classifier that depends on finding the hyper-plane which makes two given categories somewhat different. SVMs are efficient in situations where the number of dimensions exceeds the number of instances [19].

B. Deep Learning Techniques

- Deep Neural Network (DNN) : The earliest neural network models, like perceptrons, were tiny and had only one input layer, one output layer, and maybe one hidden layer in between [17]. "Deep" Learning is defined as more than three layers (input and output included). It is a notion with a restricted definition, denoting multiple hidden surfaces. Each layer of nodes in a Deep Learning network trains on a different set of features dependent on how the layers that came before it performed [15]. As nodes integrate and recombine traits from the previous layers, the properties they recognise become more complicated as you move further into the neural net.

- Recurrent Neural Network (RNN) : A RNN is a type of artificial neural network which has a "memory" that stores the necessary prior data. Hidden State, which retrieves specific information about any given sequence, such as a word set in a sentence, is the essential component of the RNNs. Several copies of the same architecture are made with RNNs, with each copy sending data to the network after it [20]. It lowers the parameter complexity in contrast to other neural networks. Although RNNs have numerous advantages, it has the vanishing gradient problem.
- Long Short-Term Memory (LSTM) : Several variants have been developed to resolve the problem of Gradients, vanishing in the RNNs. LSTM is considered to be the best of them. In theory, a repeating LSTM system tries to "remember" all past information that the network has so far seen and tries to "forget" irrelevant data [22]. To do this, additional levels of activation functions known as "gates" are added for various uses. Each recurrent LSTM unit also preserves a vector called the Internal Cell State that determines the information that the previous recurrent LSTM unit has chosen to maintain conceptually [21]. LSTM has four different gates : Input, Output, Input Modulation, and Forget Gate.
- Gated Recurrent Unit (GRU) : The goal of the GRU is to address the vanishing gradient issue that arises when using a standard recurrent neural network. The GRU is thought of as an LSTM variation [30]. They sometimes produce similarly good effects and have comparable structures. Unlike LSTM, this has just three gates and does not maintain an internal cell state. The hidden state of the Gated Recurrent Unit is filled with the data from an LSTM recurrent unit in the Inner Cell State [25]. The subsequent Gated Recurrent System will receive the shared data. A GRU has three distinct gates : Update, Reset, and Current Memory Gate [28].
- Convolutional Neural Network (CNN) : The Convolutional Neural Network (CNN), which was first created to perform Deep Learning for computer vision applications, has shown to be incredibly effective. We used the concept of a "convolution", which is a sliding window or "filter" that moves across the image, identifying and assessing each significant characteristic separately, then distilling them down to their most crucial components and repeating the process [29]. An input sentence is first divided into embedding words or words, which are low-dimensional representations made by models like GloVe or Word2Vec. Words are input into a convolutional layer based upon their properties. Either "pooled" or aggregated to a representative amount, are the convolution results. This number is fed to a neural network that is completely connected, making a classification decision based upon the weights assigned to each function within the text [18].
- Hierarchical Attention Network (HAN) : Based on the same idea as the Attention GRU, HAN was introduced in [24]. Bi-directional GRU is used in the construction of the HAN architecture to obtain the word context. It has two attention levels for each word and sentence.

IV. System Design

A. Architecture Diagram

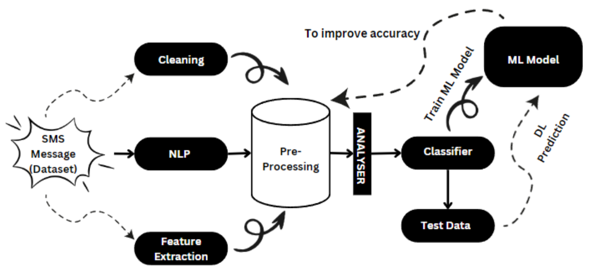

Figure 1 : Architecture Diagram

The above diagram improves the accuracy of an SMS spam classifier using Machine Learning (ML) and Deep Learning (DL). Following is the description of every component in the above Architecture Diagram :

- **SMS Messages (Dataset) :** It refers to a collection of SMS messages that will be used to train the Machine Learning model.
- **Cleaning :** The SMS messages go through a cleaning process to remove irrelevant or nonsensical information. This could include things like punctuation, special characters, or extra spaces.
- **NLP (Natural Language Processing) :** After the messages are cleaned, they are then processed using Natural Language Processing (NLP) techniques. NLPis an area of study in Computer Science that addresses the interaction between computers and human language.
- **Feature Extraction :** Once the messages have been cleaned and processed using NLP, features are extracted from them. Features are characteristics of the messages that can be used to identify spam.
- **Pre-Processing :** To prepare raw data for analysis and modelling, Pre-Processing is an essential phase in the data analysis pipeline. It involves organizing, cleaning, and transforming the data. It entails a number of steps intended to enhance the data's quality and increase its comprehensibility and use for Machine Learning algorithms and other analytical methods.

- **Analyser :** It is a component or module that handles textual data processing and analysis in Natural Language Processing (NLP). There may be tasks like sentiment analysis (determining the sentiment or emotion expressed in the text), stemming (reducing words to their root form), lemmatization (reducing words to their base or dictionary form), part-of-speech tagging (identifying the grammatical parts of words), tokenization (breaking text into individual words or tokens), and so on.
- **Classifier :** A kind of algorithm or model that labels or classifies input data into one or more predefined classes or categories; in Machine Learning and statistics. It takes a set of input features, and, based upon the patterns found in the data, gives a class label to each instance.
- **Train ML Model :** The extracted features are then used to train a Machine Learning model. The Machine Learning model is a computer program that learns to identify spam messages based upon the features that it has been trained on.
- **Test Data :** After the Machine Learning model has been trained, it is tested on a separate dataset of SMS messages. The test dataset is used to evaluate the accuracy of the model.
- **DL (Deep Learning) Prediction :** Based upon the extracted features from the SMS message, the Deep Learning model predicts whether the message is spam or not.
- **ML (Machine Learning) Model :** This refers to the Machine Learning model that has been trained to identify spam messages.

B. Data Flow Diagram

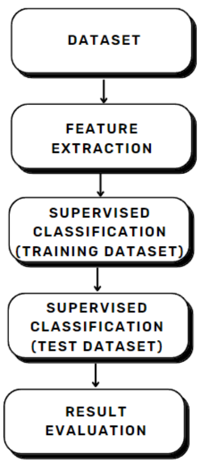

Figure 2 : Data Flow Diagram

The above diagram depicts a process of data classification, but it specifically refers to Supervised Learning for classification :

- **Dataset :** A collection of data that will be used to train the Machine Learning model. In this case, the data is likely text data.
- **Feature Extraction :** The features are characteristics of the data that can be used to classify it. In text classification, these features might be words, phrases, or other characteristics of the text that can be used to determine its category.
- **Supervised Classification (Training Dataset) :** It entails utilizing a labelled dataset to train a classification model. Each example in the dataset for Supervised Learning includes both the input features and the associated target label or class.
- **Supervised Classification (Test Dataset) :** Apply the trained model to a separate dataset to assess how well it generalizes to unseen data. In this stage, feature extraction is performed on the test dataset, and the model makes predictions about the labels of the data points in the test dataset.
- **Result Evaluation :** Evaluate the performance of the Machine Learning model on the test dataset. This typically involves metrics like accuracy, precision, and recall.

C. Use Case Diagram

Use Case diagrams identify the functionalities provided by the use cases, the actors who interact with the system, and the association between the actors and the functionalities.

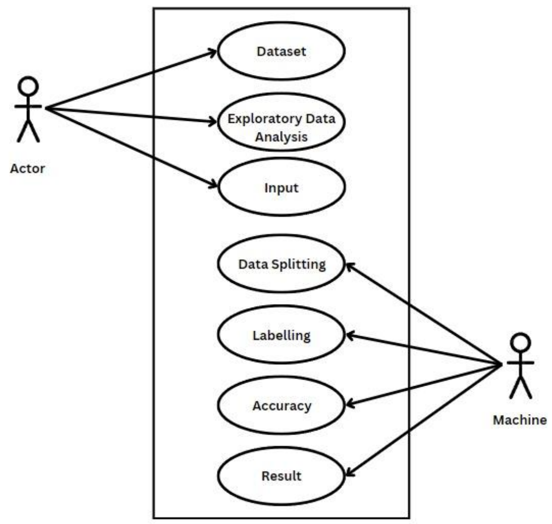

Figure 3 : Use Case Diagram

V. SOFTWARE SPECIFICATION

The Software Requirements Specification (SRS) document is the specification of the system. It should include both a definition and the specification of the requirements. It is a set of guidelines regarding what the system should do rather than how it should do it. The software requirements provide a basis for creating the Software Requirements Specification (SRS) document. It is useful in estimating costs, planning team activities, performing tasks, and tracking the teams' progress throughout the development activity.

PYTHON IDE : Anaconda Jupyter Notebook
PROGRAMMING LANGUAGE : Python

VI. RESULTS AND DISCUSSIONS

There are a few data sets to predict spam in Short Message Service (SMS) systems, among which UCI datasets are used in most studies. Thus, in this section of the paper, we first introduce this data set briefly. Then we explain the quality of performing the experiments, where the results on this data set, in classification, show satisfactory improvement.

A. Data Set Used

The dataset used in this study is of the dataset prepared in UCI, known as the SMS Spam Collection Dataset, and the dataset includes 5573 SMS, labelled and classified into two groups : 87.37 % ham and 12.63 % spam. This depicts the proportion between ham and spam drastically increases with inclusion of numbers in the text. Therefore, data is imbalanced.

B. Implementation Details

For implementation, we have used Machine Learning and Deep Learning algorithms like Multinomial NB, Decision Tree, K-Nearest Neighbor, Random Forest, AdaBoost, Gradient Boosting, Extra Trees, Bagging, XGB Classifier, LSTM, Bi-LSTM. We also created our own custom model.

Initially, we divided the data into training and test sets by converting the text label to a numeric format. Also, we converted the label to numpy arrays to fit the Deep Learning models. 80 % of the data was used for training and 20 % was used for testing purposes. As Deep Learning models do not understand text, we converted text into numerical representation. For this purpose, the first step is Tokenization. Sentences are broken up into words and then encoded into numbers using TensorFlow Keras' Tokenizer API.

After Tokenization, we used texts_to_sequences from the Tokenizer object to express each phrase as a series of numbers. Subsequently, we padded the sequence so that every sequence can be the same length. Training and assessment both use padding and sequencing.

All three models (i.e., LSTM, BiLSTM and our custom model) classify the first message ("You have won $100192") and the third message ("You should click the link below...") as spam with very high probabilities (close to 1). This indicates strong agreement on these messages being likely to be spam. All three models assign very low probabilities to the second message ("where are you?") being spam, suggesting that they correctly understand it to be ham. The custom model seems to be slightly less certain about the firstmessage compared to the LSTM and BiLSTM models, judging by a slightly lower probability (0.7758 for custom v/s near 1.0 for LSTM/BiLSTM). However, all models still classify it as highly likely to be spam. There are very minor differences for the second and the third messages between the models. All three models appear to perform well on these sample messages, effectively identifying potential spam messages.

C. Graphs

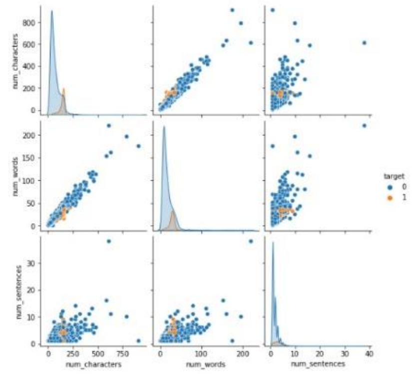

Figure 4 : Pair Plot

Using the Seaborn ('sns') library, a pair plot is shown in Figure 4. A grid of scatterplots illustrating the correlations between two pairs of variables in a DataFrame is called a Pair Plot. Every Scatterplot in the grid shows the relationship between two variables. You can see how the variables relate to the target variable by using the hue parameter, which colours the points based on values in the DataFrame's "target" column. '0' and '1' typically stand for two groups or classes. "Not spam" is represented by the number "0", whereas "spam" is represented by the number "1". The distributions of "num words", "num characters", and "num sentences" are compared in this code.

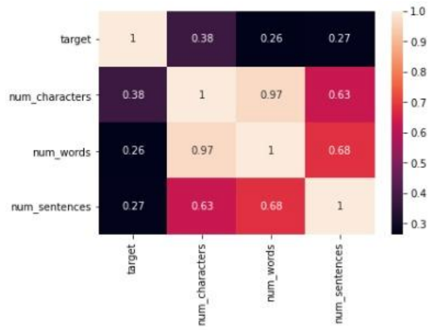

Figure 5 : Heat Map

A heatmap showing the DataFrame "df"'s correlation matrix is shown in Figure 5. From -1 to 1, the correlation coefficient is the range. A Perfect Positive Correlation is represented by a value of 1, meaning that when one variable grows, the other also increases linearly. A Perfect Negative Correlation is represented by a value of -1, meaning that as one variable rises, the other falls linearly. There is no linear association between the variables when the value is 0. Stronger positive relationships are shown by darker hues. Greater negative associations are shown by lighter hues. Shades that are nearer white indicate weaker or non-existent relationships.

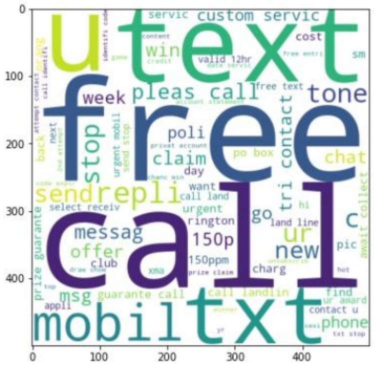

Figure 6 : Cloud Map (Spam)

Figure 6 (plt.figure(figsize=(15,6))) shows a specific figure that is 15 by 6 inches. Next, it shows an image that is represented by the variable spam_wc using plt.imshow(). Words that appear more frequently in the text data are rendered larger in the image, creating a word cloud related to spam messages. Word clouds are widely used to visualize text data and quickly identify phrases that appear frequently.

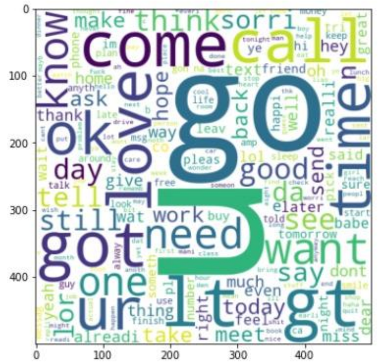

Figure 7 : Cloud Map (Ham)

Figure 7 (plt.figure(figsize=(15,6))) shows a specific figure that is 15 by 6 inches. Next, it shows an image that is represented by the variable ham_wc using plt.imshow(). Words that appear more frequently in the text data are rendered larger in the image, creating a word cloud related to ham messages.

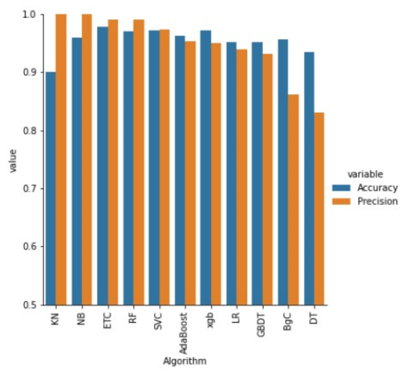

Figure 8 : Comparison b/w Machine Learning (ML) algorithms

Figure 8 compares the accuracy and precision of 11 Machine Learning algorithms in classifying messages into spam and ham.

VII. Comparative Analysis

We evaluated and contrasted various Machine Learning algorithms based upon their accuracy and precision. The KNN algorithm has an overall precision of 1.000000 and an accuracy of 0.900387. Comparatively, the BGC algorithm has a low overall precision (0.861538) but a good overall accuracy (0.957447). This indicates that while the BGC algorithm frequently makes accurate predictions, they are not always very consistent with each other. After evaluating every Machine Learning algorithm, it was determined that whereas AdaBoost, Linear Regression, Support Vector Machines, and KNN algorithms have high precision and high accuracy; Random Forest, ET, and XGB methods have high precision but low accuracy. Additionally, we tested our own custom model against both LSTM and Bi-LSTM, and discovered that all three models appear to be able to distinguish between messages that are spam and those that are not. While the performance of the custom model is respectable, it lacks the sophistication of the LSTM and the Bi-directional LSTM models. In certain cases, the Bi-directional LSTM model performs slightly better than the LSTM model because it gives spam messages with slightly greater probabilities.

VIII. Conclusion

Nearly every nation is plagued by the SMS spam issue, which is growing and shows no signs of abating as the number of mobile users increases in addition to cheap rates of SMS services. Therefore, this paper presents the Spam Filtering technique using various Machine Learning algorithms and Deep Learning techniques. Different algorithms will perform differently and provide different results based upon the features used. For future works, adding more characteristics, like message durations, could aid the classifiers in training data better and giving better performance.

IX. Future Scope

The future scope of this project will involve adding more feature parameters. More the parameters taken into account, higher will be the accuracy. The algorithms can also be applied to analyze the content of public comments and thus determine patterns/relationships between the customer and the company. The use of traditional algorithms and data mining techniques can also help predict the corporation's performance structure as a whole.

X. Applications

- Can be used by companies to prevent users from using fake links.
- Hacking can be prevented.

Acknowledgement

It is our pleasure to express with deep sense of gratitude to Dr. Padmanaban R. of Vellore Institute of Technology, Chennai for his constant guidance, kind encouragement in doing this project and completing it successfully.

Our sincere thanks to the VIT, Chennai Management for providing us with the necessary support and details timely during the progressive reviews.

We wish to express our gratitude to all the Teaching and Non-Teaching staff members of the Department of Computer Science and Engineering (SCOPE), who were helpful in many ways for the completion of the project.

REFERENCES

[1] SMS Spam Detection using Machine Learning and Deep Learning, Sridevi Gadde, 2021.

[2] Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia, Agustinus Theodorus, Tio Kristian Prasetyo, Reynaldi Hartono, and Derwin Suhartono, 2021.

[3] Spam Detection Approach for Secure Mobile Message Communication using Machine Learning Algorithms, Shah Nazir, Habib Ullah Khan, and Amin Ul Haq, 2020.

[4] The Impact of Deep Learning Techniques on SMS Spam Filtering, Wael Hassan Gomaa, 2020.

[5] SMS Spam Detection using Machine Learning Approach, Houshmand Shirani-Mehr, 2019.

[6] Comparative Study of Machine Learning Algorithms for SMS Spam Detection, Amani Alzahrani, and Danda B. Rawat, 2019.

[7] SMS Spam Detection based on Long Short-Term Memory and Gated Recurrent Unit, Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop, 2019.

[8] A Review of Soft Techniques for SMS Spam Classification : Methods, Approaches and Applications, Olusola Abayomi-Alli, Sanjay Misra, Adebayo Abayomi-Alli, and Modupe Odusami, 2019.

[9] SMS Spam Filtering using Supervised Machine Learning Algorithms, Pavas Navaney, Gaurav Dubey, and Ajay Rana, 2018

[10] A Review on Mobile SMS Spam Filtering Techniques, Shafi'l Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwasfemi Osho, Gaddafi Adbul-Salaam, and Adamu I. Abub, 2017.

[11] Towards Filtering of SMS Spam Messages using Machine Learning-based Techniques, Neelam Choudhary, and Ankit Kumar Jain, 2017.

[12] Spam Filtering in SMS using Recurrent Neural Networks, Rahim Taheri, and Reza Javidan, 2017.

[13] SMS Spam Filtering for Modern Mobile Devices, N. A Azeez, and O. Mbaike, 2017.

[14] SMS Spam Filtering and Threat Identification using Bi-level Text Classification and Clustering Techniques, Naresh Kumar Nagwani, and Aakanksha Sharaff, 2016.

[15] Factorial Design Analysis applied to the Performance of SMS Anti-Spam Filtering Systems, Marcelo V. C. Argao, Edielson Prevato Frigieri, Carlos A. Ynoguti, and Anderson P. Paiva, 2016.

[16] Text Normalization and Semantic Indexing to Enhance Instant Messaging and SMS Spam Filtering, Tiago A. Almeida, Tiago P. Silva, Igor Santos, and Jose M. Gomez Hidalgo, 2016.

[17] Intelligent SMS Spam Filtering using Topic Model, Jialin Ma, Yongjun Zhang, Jinling Liu, Kun Yu, and XuAn Wang, 2016.

[18] A Method of SMS Spam Filtering based on AdaBoost Algorithm, Xipeng Zhang, Gang Xiong*, Yuexiang Hu, Fenghua Zhu, Xisong Dong, and Timo R. Nyberg, 2016.

[19] Metin Sınıflandırma ve Uzman Sistem Tabanlı ˙Istenmeyen Kısa Mesajların Filtrelenmesi SMS Spam Filtering based on Text Classification and Expert System, Yavuz Selim Bozan, Önder Çoban, Gül¸sah Tümüklü Özyer, and Barı¸s Özyer, 2015.

[20] SMS Spam Filtering Application using Android, Gaurav Sethi, and Vijender Bhootna, 2014.

[21] The Impact of Feature Extraction and Selection on SMS Spam Filtering, A. K. Uysal, S.Gunal, S.Ergin, and E. Sora Gunal, 2013.

[22] Content-based Hybrid SMS Spam Filtering System, T. Chaminda, T. T. Dayaratne, H. K. N. Amarasinghe, and J. M. R. S. Jayakody, 2013.

[23] Mobile SMS Spam Filtering for Nepali Text using Naïve-Bayesian and Support Vector Machine, Tej Bahadur Shahi, and Abhimanu Yadav, 2013.

[24] SMS Spam Filtering : Methods and Data, Sarah Jane Delany, Mark Buckley, and Derek Greene, 2012.

[25] SMS Spam Filtering Technique based on Artificial Immune System, Tarek M Mahmoud, and Ahmed M Mahfouz, 2012.

[26] Simple SMS Spam Filtering on Independent Mobile Phone, M.Taufiq Nuruzzaman, Chanmoo Lee, Mohd. Fikri Azli bin Abdullah, and Deokjai Choi, 2012.

[27] A Novel Framework for SMS Spam Filtering, Alper Kursat Uysal, Serkan Gunal, Semih Ergin, and Efnan Sora Gunal, 2012.

[28] Independent and Personal SMS Spam Filtering, M. Taufiq Nuruzzaman, Changmoo Lee, and Deokjai Choi, 2011.

[29] SMSAssassin : Crowdsourcing Driven Mobile-based System for SMS Spam Filtering, Kuldeep Yadav, Ponnurangam Kumaraguru, Atul Goyal, Ashish Gupta, and Vinayak Naik, 2011.

[30] Spam Filtering for Short Messages, Gordon V. Cormack, Jose Maria Gomez, and Enrique Puertas Sanz, 2007.