SMS SPAM DETECTOR USING MACHINE LEARNING

ELUMALAI S[1], GOKUL J[2], HARISH K[3] , MADHUMATHI S [4]

[1,2,3] *UG Scholar, Department of CSE, Kingston College, Vellore-59*

[4] *Asst.Professor, Department of CSE, Kingston College, Vellore-59*

---***---

ABSTRACT

Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, a reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. In parts of Asia, up to 30% of text messages were spam in 2012. The lack of real archive for SMS mailshots, a economize of messages, gallop features, and their demonic language are the factors that may cause the established email filtering algorithms to underperform in their classification. In this predict, a database of authentic SMS Spam from the UCI Machine Learning repository is used, and after pre-processing and feature extraction, different machine learning techniques are applied to the database. Finally, the results are compared and the best algorithm for spam filtering for text messaging is introduced. Final simulation solution using 10-fold cross-validation show the best unified in this work force into the overall error rate of the best model in the original paper citing this dataset by more than half. Algorithms used in this technique are: Logistic regression (LR), K-nearest neighbor (K-NN), and Decision tree (DT) are used for the classification of spam messages in mobile device communication. The SMS spam collection set is used for testing the method.

Key Words**:** Logistic regression(LR) algorithm and K-nearest neighbor(K-NN) and Decision tree (DT).

1.INTRODUCTION

Short Message Services is a lot more than just a technology for a converse. message technology move forward out of the global system for mobile communications standard, an internationally accepted. Spam is the abuse of electronic messaging systems to send unsolicited messages in bulk indiscriminately. While the most acknowledged form of spam is email spam, the term is applied to similar abuses in other media and mediums. SMS Spam in the context is very similar to email spams, typically, unsolicited bulk messaging with some business interest. SMS spam is used for commercial advertising and spreading exploit links. Commercial spammers use malware to send message spam because sending SMS spam is illegal in most great outdoors. Sending spam from a compromised machine reduces the risk to the spammer because it obscures the derivation of the spam. Messages can have a limited number of characters, which take in alphabets, numbers, and a few symbols. A inspect the messages shows a distinct pattern. Almost all of the spam messages ask the users to call a number, reply by SMS, or visit some URL. This pattern is observable by the results obtained by a simple SQL query on the spam. The low price and the high bandwidth of the SMS network have attracted a large amount of SMS spam.

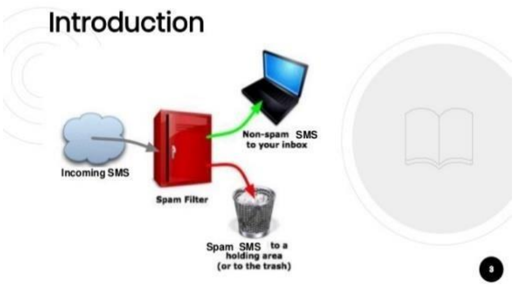

Introduction
Incoming SMS
Spam Filter
Non-spam SMS to your inbox
Spam SMS to a holding area (or to the trash)

Figure : INTRODUCTION

2. RELATED WORKS

[1] Zhou, Y.F., Wang, Z.M.: Dynamic instability of axially moving viscoelastic plate. Eur. J. Mech. A Solids 73, 1–10 (2019)

This paper is devoted to the investigation of the transverse vibration and dynamic stability of the axially moving viscoelastic plate with two opposite edges simply supported and other two opposite edges with simply supported or free. By considering the Kelvin–Voigt model of viscoelasticity, the equation of motion of the plate is derived. The normalized power series method is employed to obtain the complex eigen equations for the axially moving viscoelastic plate. The variation relationship between the first three complex frequencies of the system and the dimensionless axially moving speed with different aspect ratio and dimensionless delay time are analyzed. The results show that the dimensionless delay time, axially moving speed as well as the aspect ratio have remarkable effects on dynamic behaviors and stability of the axially moving viscoelastic plate.

[2] K.-S. Hong and P.-T. Pham, "Control of axially moving systems: A review," Int. J. Control Autom. Syst., vol. 17, no. 12, pp. 2983–3008, 2019.

A comprehensive review of significant works on active vibration control of axially moving systems. Owing to their broad applications, vibration suppression techniques for these systems have generated active research over decades. Mathematical equations for five different models (i.e., string, beam, coupled, plate, and approximated model) are outlined. Active vibration control of axially moving systems can be performed based on a finite-dimensional model described by ordinary differential equations (ODEs) or an infinite-dimensional model described by partial differential equations (PDEs). For ODE models, the sliding mode control is most representative. For PDE models, however, there exist various methods, including wave cancellation, Lyapunov method, adaptive control, and hybrid control. Control applications (lifting systems, steel industry, flexible electronics, and roll-to-roll systems) are also illustrated. Finally, several issues for future research in vibration control of axially moving systems are discussed.

[3] Z. Zhao and Z. Li, "Finite-time convergence disturbance rejection control for a flexible Timoshenko manipulator," IEEE/CAA J. Automatica Sinica, early access, 2020

A new finite-time convergence disturbance rejection control scheme design for a flexible Timoshenko manipulator subject to extraneous disturbances. To suppress the shear deformation and elastic oscillation, position the manipulator in a desired angle, and ensure the finitetime convergence of disturbances, we develop three disturbance observers (DOs) and boundary controllers. Under the derived DOs-based control schemes, the controlled system is guaranteed to be uniformly bounded stable and disturbance estimation errors converge to zero in a finite time. In the end, numerical simulations are established by finite difference methods to demonstrate the effectiveness of the devised scheme by selecting appropriate parameters.

PROPOSED SYSTEM:

Although there are various SMS junk dripple performance available, still there is a need to handle this problem with advanced skills. spam messages can be two types junk mail. The purpose of email junk or SMS spam is the identical. Generally, these spam messages are spent by spammers for the promotion of their utilities or business. Sometimes, the users may also undergo financial loss due to these spam messages. Machine Learning is a technology, where machines learn from previous data and made a soothsaying on future data. Nowadays, machine learning and deep learning can be applied to solve most of the physical world problems in all sectors like health, security, market analysis, etc. There are types of machine learning like supervised learning, unsupervised, oppress learning, etc. In supervised learning, the dataset is having output labels, whereas non-self learning deals with database with no labels. We used a dataset from UCI with labels, So we applied various supervised learning algorithms for SMS spam detection.

Advantages :

- These models are developed using the same dataset with our proposed models.
- The LSTM and GRU algorithms have a high-yielding than the SVM and NB algorithms.
- From these results, it might be make know that the deep learning algorithms provide a better performance than the models developed based on model based-machine learning algorithms for SMS spam classification in English language.

4.3.1.ALGORITHMS:

I. K-Nearest Neighbours

k-nearest neighbour can be applied to the classification problems as a simple instance-based learning algorithm. In this method, the label for a test sample is predicted based on the majority vote of its k nearest neighbours.

Kernel Function	Overall Error %	Spam Caught (SC) %	Blocked Hams (BH) %
Linear	1.18	93.8	0.47
Degree-2 Polynomial	2.03	85.7	0.27
Degree-3 Polynomial	1.64	89.7	0.4.
Degree-4 Polynomial	1.70	90.65	0.6

TABLE:10-fold cross validation error of k-nearest neighbour classifier

II. Support Vector Machines (SVM)

In support vector machine is applied to the dataset. Table II shows the 10-fold cross validation results of SVM with different kernels applied to the dataset with extracted features. As it is shown in the table, linear kernel gains better performance compared to other mappings. Using the polynomial kernel and increasing the degree of the polynomial from two to three shows improvement in error rates, however the error rate does not improve when the degree is increased further. Radial basis function (RBF) is another kernel applied here to the dataset. RBF kernel on two samples x1 and x2 is expressed by following equation:

$K(x1, x2) = \exp(- kx1 - x2k\ 2\ 2\ 2\sigma2\ )$

K	Overall error %	Spam Caught (SC) %	Blocked Hams (BH) %
2	2.78	81.3	0.46
10	2.53	82.6	0.40
20	2.98	78.8	0.35
50	3.4	74.8	0.24
100	4.14	68.4	0.16

TABLE: 10-fold cross validation error of SVM with different kernel functions on dataset

Radial Basis Function	2.61	81.45	0.32
Sigmoid	13.4	0	0

From the analysis of results, we notice that the length of the text message (number of characters used) is a very good feature for the classification of spams. Sorting features based on their mutual information (MI) criteria shows that this feature has the highest MI with target labels.

Additionally, going through the misclassified samples, we notice that text messages with length below a certain threshold are usually hams, yet because of the tokens corresponding to the alphabetic words or numeric strings in the message they might be classified as spams.

While applying SVM with different kernels increases the complexity of the model and subsequently the running time of training the model on data, the results show no benefit compared to the multinomial naive Bayes algorithm in terms of accuracy.

III. Random Forest

Random forests is an equating ensemble method for classification. The combo is a combination of decision trees built from a reset sample from training set. Additionally, in building the decision tree, the split which is chosen when splitting a node is the best split only among a random set of features. This will increase the bias of a single model, but the averaging reduces the variance and can compensate for increase in bias too. Consequently, a better model is built. In this work, the implementation of random forests in scikitlearn python library is used, which averages the probabilistic predictions. Two number of estimators are simulated for this method. With 10 estimators, the overall error is 2.16%, SC is 87.7 %, and BH is 0.73%. Using 100 estimators will result in overall error of 1.41 %, SC of 92.2 %, and BH of 0.51 %. We observe that comparing to the naive Bayes algorithm, although the complexity of the model is increased, yet the performance does not show any improvement.

IV. Naive Bayes Theorem

Step 1: Introduction to the Naive Bayes Theorem

Bayes theorem is one of the too early probabilistic conjecture algorithms developed by Reverend Bayes (which he used to try and infer the extant of God no less) and still performs extremely well for established use cases. It's finest to understand this theorem using an example. Let's say you are a member of the unrevealed Service and you have been deployed to protect the Democratic presidential nominee during one of his/her campaign speeches. Being a public event that is open to all, your job is not easy and you have to be on the persistent lookout for threats. So one place to start is to put a definite threat-factor for each person. So based on the attribute of an individual, like the age, sex, and other smaller factors like is the person carrying a bag?, does the person look easily frightened? etc. you can make a judgement call as to if that person is feasible threat. If an individual stroke all the boxes up to a level where it crosses a threshold of doubt in your mind, you can take action and remove that person from the vicinity. The Bayes theorem works in the same way as we are computing the likeliness of an event(a person being a threat) based on the probabilities of fixed coupled events(age, sex, presence of bag or not, nervousness etc. of the person).

Step 2: comprehension our dataset

We will be using a dataset from the UCI Machine Learning repository which has a very good collection of datasets for experimental research purposes.

Ham Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine ther Ham Ok lar... Joking wif u oni...

Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 t

Spam 08452810075over18's

Ham U dun say so early hor... U c already then say...

Ham Nah I don't think he goes to usf, he lives around here though

FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun yo

Spam to rcv

Ham Even my brother is not like to speak with me. They treat me like aids patent.

Ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set WINNER!! As a valued network customer you have been selected to receivea å£900 priz

Spam only.

Spam Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles Ham I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575.

Spam info

URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Spam 4403LDNW1A7RW18

Ham I've been searching for the right words to thank you for this breather. I promise i wont tak Ham I HAVE A DATE ON SUNDAY WITH WILL!!

XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message o

Spam

Step 3: Data Preprocessing

Now that we have a basic understanding of what our dataset looks like, lets convert our labels to binary variables, 0 to represent 'ham'(i.e. not spam) and 1 to represent 'spam' for ease of computation. You might be wondering why do we need to do this step? The answer to this lies in how scikit-learn handles inputs. Scikit-learn only deals with numerical values and hence if we were to leave our label values as strings, scikit-learn would do the conversion internally(more specifically, the string labels will be cast to unknown float values). Our model would still be able to make predictions if we left our labels as strings but we could have issues later when calculating performance metrics, for example when calculating our precision and recall scores. Hence, to avoid unexpected 'gotchas' later, it is good practice to have our categorical values be fed into our model as integersxxxmobilemovieclub.com?n=QJKGIGHJJ GCBL

Ham Oh k...i'm watching here:)

The columns in the data set are currently not named and as you can see, there are 2 columns.

The first column takes two values, 'ham' which signifies that the message is not spam, and 'spam' which signifies that the message is spam.

The second column is the text content of the SMS message that is being classified.

Step 4: Bag of words

What we have here in our data set is a large group of text data (5,572 rows of data). Most ML algorithms rely on numerical data to be fed into them as input, and email/sms messages are usually text heavy. Here we'd like to introduce the Bag of Words (BOW) concept which is a term used to specify the problems that have a 'bag of words' or a collection of text data that needs to be worked with. The basic idea of BOW is to take a piece of text and count the frequency of the words in that text. It is important to note that the BOW concept treats each word individually and the order in which the words occur does not matter. Using a process which we will go through now, we can convert a collection of documents to a matrix, with each document being a row and each word(token) being the column, and the corresponding (row, column) values being the frequency of occurrence of each word or token in that document. Data pre-processing with CountVectorizer() Some of important parameters of CountVectorizer().

1. lowercase = True The lowercase parameter has a default value of True which converts all of our text to its lowercase form. 2. Token pattern = (?u)\b\w\w+\b The token pattern parameter has a default regular expression value of (?u)\b\w\w+\b which ignores all punctuation marks and treats them as delimiters, while accepting alphanumeric strings of length greater than or equal to 2, as individual tokens or words.

3. Stop words The stop words parameter, if set to english will remove all words from our document set that match a list of English stop words which is defined in scikit-learn. Considering the size of our dataset and the fact that we are dealing with SMS messages and not larger text sources like e-mail, we will not be setting this parameter value.

Step 5: Training and trial sets

Now that we have understood how to deal with the Bag of Words problem we can get back to our dataset and proceed with our analysis. Our first step in this regard would be to split our dataset into a training and testing set so we can test our model later.

Instructions: Split the dataset into a training and testing set by using the train_test_split method in sklearn. Split the data using the following variables:

- X__train is our training data for the 'sms_message' column.
- Y_train is our training data for the 'label' column
- X_test is our testing data for the 'sms_message' column.
- y_test is our testing data for the 'label' column Print out the number of rows we have in each our training and testing data.

Step 6: Applying Bag of Words processing to our dataset

Now that we have split the data, our next objective is to follow the steps from

Bag of words and convert our data into the selected matrix format. To do this we will be using CountVectorizer() as we did before. There are two steps to consider here:

- Firstly, we have to fit our training data (X_train) into CountVectorizer() and return the matrix.
- Secondly, we have to transform our testing data (X_test) to return the matrix. Note that X_train is our training data for the 'sms_message' column in our dataset and we will be using this to train our model. X_test is our testing data for the 'sms_message' column and this is the data we will be using(after transformation to a matrix) to make predictions on. We will then compare those predictions with y_test in a later step.

Step-7: Implementation of Naive Bayes Machine Learning Algorithm

I will use sklearns sklearn.naive_bayes approach to make predictions on our dataset for SMS Spam Detection.

Specifically, we can be the use of the multinomial Naive Bayes implementation. This unique classifier is appropriate for class with discrete features. It takes in integer phrase counts as its input.

Predictions
=naive_bayes.predict(testing_data)

Step 8: Evaluating our version

Now that we've made predictions on our check set, our subsequent purpose is to assess how properly our version is doing. There are numerous mechanisms for doing so, however first let's do short recap of them.

Accuracy measures how frequently the classifier makes the precise prediction. It’s the ratio of the wide variety of accurate predictions to the overall wide variety of predictions (the wide variety of check statistics points).

Precision tells us what share of messages we categorized as unsolicited mail, absolutely have been unsolicited mail. It is a ratio of genuine positives(phrases categorized as unsolicited mail, and which can be absolutely unsolicited mail) to all positives(all phrases categorized as unsolicited mail, regardless of whether or not that changed into the precise class), in different phrases it's miles the ratio of True Positives/(True Positives + False Positives)

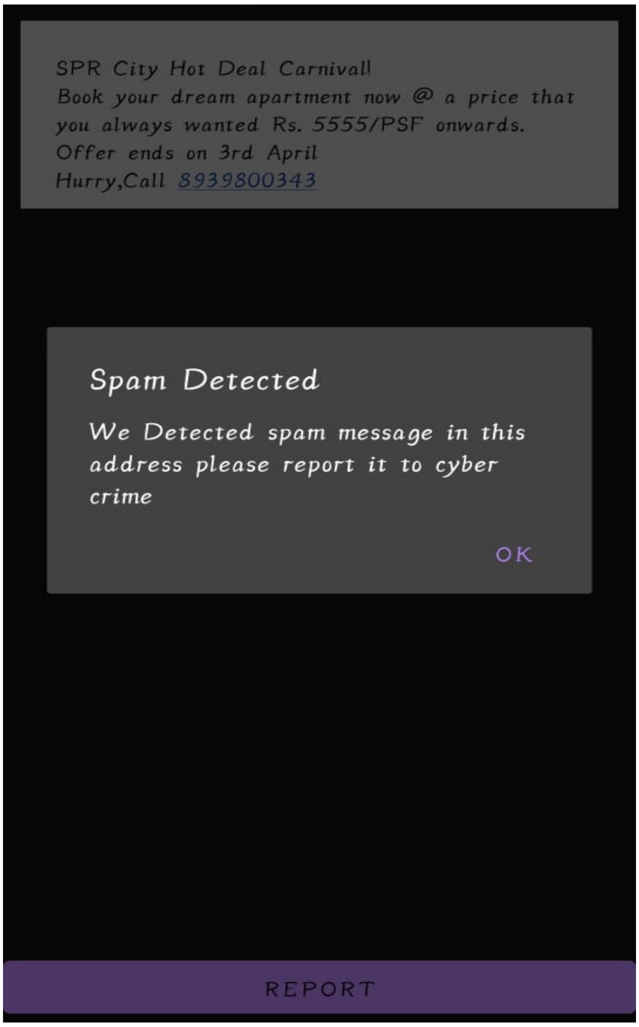
Recall(sensitivity) tells us what share of messages that absolutely have been unsolicited mail have been categorized through us as unsolicited mail. It is a ratio of genuine positives(phrases categorized as unsolicited mail, and which can be absolutely unsolicited mail) to all of the phrases that have been absolutely unsolicited mail, in different phrases it's miles the ratio of

True Positives/(True Positives + False Negatives)

For class issues which might be skewed of their class distributions like in our case, as an example if we had a a hundred textual content messages and simplest 2 have been unsolicited mail and the relaxation 98.

Conclusion

One of the major advantages that Naïve Bayes has over other classification algorithms is its ability to handle an extremely large number of features. In our case, each word is treated as a feature and there are thousands of different words. Also, it performs well even with the presence of irrelevant features and is relatively unaffected by them. The other major advantage it has is its relative simplicity. Naïve Bayes' works well right out of the box and tuning it's parameters is rarely ever necessary, except usually in cases where the distribution of the data is known. It rarely ever overfits the data. Another important advantage is that its model training and prediction times are very fast for the amount of data it can handle. All in all, Naïve Bayes' really is a gem of an algorithm!



SPR City Hot Deal Carnival!
Book your dream apartment now @ a price that you always wanted Rs. 5555/PSF onwards.
Offer ends on 3rd April
Hurry,Call 8939800343

Spam Detected

We Detected spam message in this address please report it to cyber crime

V. AdaBoost with Decision Tree

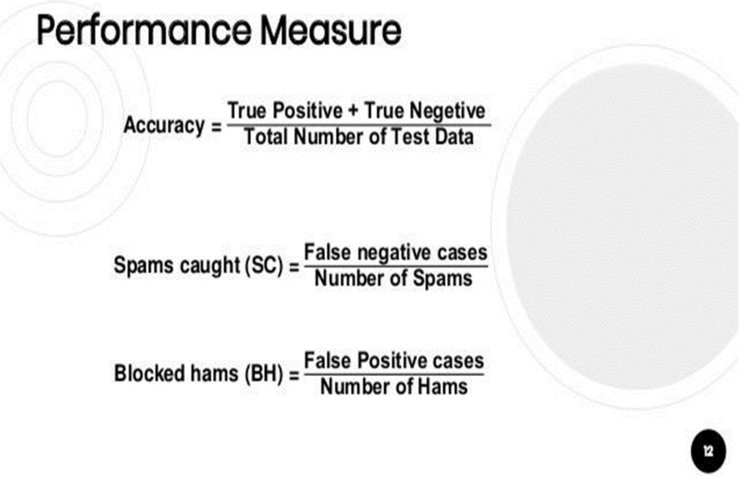
AdaBoost is a boosting ensemble method which sequentially builds classifiers that are modified in favour of misclassified instances by previous classifiers .

The classifiers it uses can be as weak as only slightly better than random guessing, and they will still improve the final model. This method can be used in conjunction with other methods to improve the final ensemble model. In each iteration of Ada Boost, certain weights are applied to training samples. These weights are distributed uniformly before first iteration. Then after each iteration, weights for misclassified labels by current model are increased, and weights for correctly classified samples are decreased. This means the new predictor focuses on weaknesses of previous classifier.

Model	SC%	BH%	Accuracy %
Multinomial NB	94.47	0.51	98.88
SVM	92.99	0.31	98.86
K-Nearest neighbour	82.60	0.40	97.47
Random Forest	90.62	0.29	98.57
Ada boost withDecision tree	92.17	0.51	98.59

Final results of different classifiers applied to SMS Spam dataset

We tried applying of Ada boost with decision trees using scikit-learn library. Using 10 estimators, the simulation shows 2.1% overall error rate, 87.7% SC, and 0.74% BH. Increasing the number of estimators to 100 will change these values to 1.41%, 92.2%, and 0.51% respectively. Like Random Forests, although the complexity is much higher, naive Bayes algorithm still beats Ada boost with decision trees in terms of performance.



Performance Measure

$$\text{Accuracy} = \frac{\text{True Positive + True Negetive}}{\text{Total Number of Test Data}}$$

$$\text{Spams caught (SC)} = \frac{\text{False negative cases}}{\text{Number of Spams}}$$

$$\text{Blocked hams (BH)} = \frac{\text{False Positive cases}}{\text{Number of Hams}}$$

References:

[1]https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. published by S.O'Dea,Feb 23,2022

[2]S. M. Abdulhamid, M. S. Abd Latif and Haruna Chiroma, "Robust Heart Disease Prediction A Review on Mobile SMS Spam Filtering Techniques", EEE Access, vol. 5, pp. 15650-15666, 2017.

[3]Nilam Nur Amir Sjarif, N F Mohd Azmi and Suriayati Chuprat, "SMS Spam Message Detection using Term Frequenct-Inverse Document Frequency and Random Forest Algorithm", The Fifth Information Systems International Conference 2019 Procedia Computer Science, vol. 161, pp. 509-515, 2019.

[4]A. Lakshmanarao, K. Chandra Sekhar and Y. Swathi, "An Efficient Spam Classification System Using Ensemble Machine Learning Algorithm", Journal of Applied Science and Computations, vol. 5, no. 9, September 2018.

[5] Luo GuangJun, Shah Nazir, Habib Ullah Khan and Amin Ul Haq, "Spam Detection Approach for Secure Mobile Messgae Communication using Machine Learning Algorithms", Hindawi Security and Communication Netwroks, vol. 2020, July 2020.

[6]Gomatham Sai Sravya, G Pradeepini and Vaddeswaram, ": Mobile Sms Spam Filter Techniques Using Machine Learning Techniques", Internatıonal Journal Of Scıentıfic & Technology Research, vol. 9, no. 03, March 2020.

[7]M. Rubin Julis and S. Alagesan, "Spam Detection In Sms Using Machine Learning through Textmining", Internatıonal Journal Of Scıentıfıc & Technology Research, vol. 9, no. 02, February 2020.

[8]K. Sree Ram Murthy, K. Kranthi Kumar, K. Srikar, C H. Nithya and S. Alagesan, "SMS Spam Detection using RNN", International Research Journal of Engineering and Technology (IRJET), vol. 07, no. 05, May 2020.

[9]S. Sheikhi, M. T. Kheirabadi and A. Bazzazi, "An Effective Model for SMS Spam Detection Using Content-based Features and Neural Network", International Journal of Engineering IJE TRANSACTIONS B: Applications, vol. 33, no. 2, pp. 221-228, February 2020.

[10] M.Rubin Julis, S.Alagesan,"Spam Detection In Sms Using Machine Learning

Through Text Mining",international journal of scientific & technology research volume 9, issue 02, february 2020 issn 2277-8616