Wireless Personal Communications (2020) 110:403–422
https://doi.org/10.1007/s11277-019-06734-y

Predicting Spam Messages Using Back Propagation Neural Network

Ankit Kumar Jain[1] · Diksha Goel[2] · Sanjli Agarwal[1] · Yukta Singh[1] · Gaurav Bajaj[1]

Published online: 14 September 2019

Abstract
With the increase in popularity of smartphones, text-based communication has also gained popularity. Availability of messaging services at low cost has resulted into the increase in spam messages. This increase in number of spam messages has become an important issue these days. Many mobile applications are developed to detect spam messages in mobile phones but still, there is a lack of a complete solution. This paper presents an approach for the detection of spam messages. We have identified an effective feature set for text messages which classify the messages into spam or ham with high accuracy. The feature selection procedure is implemented on normalized text messages to obtain a feature vector for each message. The feature vector obtained is tested on a set of machine learning algorithms to observe their efficiency. This paper also presents a comparative analysis of different algorithms on which the features are implemented. In addition, it presents the contribution of different features in spam detection. After implementation and as per the set of features selected, Artificial Neural Network Algorithm using Back Propagation technique works in the most efficient manner.

Keywords Spam messages · Feature selection · Text normalization · Text classification · Machine learning · Neural network

1 Introduction

Now a days, mobile phones have become an essential part of communication for a large number of people. More than half of population of the world is using mobile phones due to a wide range of functionalities these devices provide [1]. Ever since this technology has evolved, one of its significant features that is very popular is text messaging. This internet free service has become an important means of communication for people but one major disadvantage of this technology is the associated degree forbidding rate of spam messages that are sent over the snetwork to a large number of people.

✉ Ankit Kumar Jain
ankitjain@nitkkr.ac.in

[1] National Institute of Technology, Kurukshetra, Haryana, India

[2] The University of Adelaide, Adelaide, Australia

A report by Kaspersky [2] gave an insight on how spam messages divert the users to different https which can be fake payment gateways, Punycode encoding, fake crypto currency wallets, social media frauds or tax refunds etc. They also lists the name of the countries that rank high in getting trapped by cases of spam messages. Figure 1 shows top countries currently suffering from heavy spam traffic. To know more about the problem, a survey was conducted to know about the reaction of people towards the spam messages they are receiving. A large number of people admitted about receiving fake messages from banks containing URLs which led to vindictive downloads or links to sites which ask for personal information.

According to IBM's X-Force researchers [3], the number of spam messages are increasing at an alarming rate. For many companies, that increase is reinforcing the realization that spam is not just a mere nuisance, it is one of the primary channel for attacks, and therefore a direct threat to their organization. SMS phishing (i.e. Smishing) subset of spam messages aims at breaching the personal information of the user through SMS or text messages. Smishing attack is not only an issue of annoyance that disturbs a large population but also a matter of great dilemma that leads to financial frauds, contravention of personal information, and installation of a malware application [4].

The aim of this paper is to design a technique that can effectively detect spam messages with high accuracy. It works by identifying features that can be used to categorize messages as spam or ham message. Features selection procedure involves incorporation of those part, attributes and characteristics of the message that particularly helps in segregation of the two messages. For instance, the probability of presence of a URL (Uniform Resource Locator) is higher in case of a spam message than that of ham message. Hence, this can work as a feature for detecting spam messages. Similarly, this paper examines other features, which lift up the process of spam detection. The language used in text messages is often informal and contains a lot of abbreviations and elongated words due to which it becomes very difficult to select the features from the raw message. To address this problem, we have first processed the messages under text normalization phase that tries to build up a structure in the informal text and highlight its important characteristics [5].

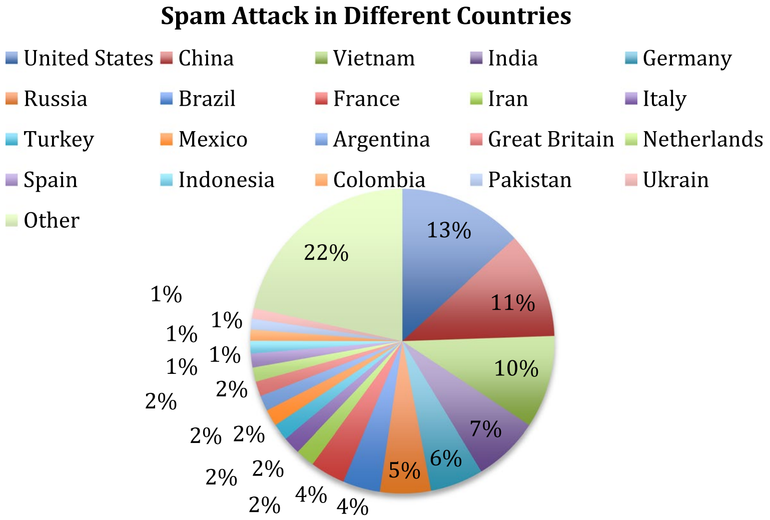


Fig. 1 Spam attacks in different countries [2]

However, during text normalization, some features like the length of the original raw messages are altered which might be a significant characteristic in determining the legitimacy of the message. Therefore, both the raw message and the processed message are used for building the feature set.

After developing a set of features, we implement a neural network on these features to determine the accuracy of the system. The flowchart in Fig. 2 shows the basic mechanism of the proposed approach.

The rest of the paper is divided into sections which examine different parts of this paper. Section 2 discusses the related work that has been done in this field. Section 3 describes the proposed approach for the detection of spam messages. This describes the detailed process of text normalization and discusses the features that build up the spam detection system. Section 4 portrays the evaluation and comparative analysis. Finally, Sect. 5 concludes the paper.

2 Related Work

Joo et al. [6] proposed an approach that uses Naïve Bayesian classifier. The classification is done on the basis of the probability of the existence of different words in the text messages of the dataset. It uses a statistical learning method. This paper considers the Naïve Bayes ham and spam probability of the message, as features in the vector generated. Etaiwi et al. [7] investigated the comparison of two features implementing them separately on the same database. The two features used in this paper include a bag of words and word count, which are compared using various algorithms like SVM, random forests, etc. However, approach achieved only 80–90% detection accuracy in the case of a bag of words and 60–70% for word count. We have tried to incorporate a better feature vector by using more features.

Patel et al. [8] proposed opinion spam detection approach that uses feature selection. The three characteristics that are worked upon include Boolean, bag-of-words and Term frequency-Inverse Document Frequency (TF-IDF). These three characteristics are implemented as features on the Naive Bayes algorithms. This works more efficiently than work of Etaiwi et al. [7], but still needs improvement to be done as a practical application. Ali et al. [9] developed a NET library written in C# which provides a cross-platform solution for spam detection. They have discussed some algorithms and have implemented Random Forests. The .NET library developed can run on any platform as per requirements. They have trained the C# library efficiently resulting in a good accuracy.

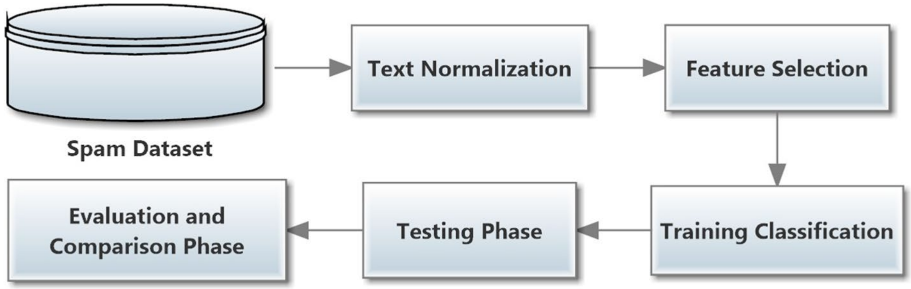


Fig. 2 General flow of the paper

Jain et al. [10] proposed a framework for the detection of smishing messages. It is a rule-based framework that discusses various rules to classify the message as spam and ham. A set of 9 characteristics like URL presence, mathematical operations etc. are considered as deciding factors for the classification of the message. Standardization of text message was performed in order to obtained better features. Wu et al. [11] proposed a mechanism based on deep learning techniques to detect spam messages on twitter. The sentence structure of each tweet will be learned through Word Vector Training Mode. A binary classifier is constructed based on the preceding representation dataset. In experiments, they worked on detecting spam in Twitter messages by observing 10-day real data from Twitter. The method largely outperformed existing methods.

Sethi et al. [12] proposed a mechanism for detecting spam messages sent to mobile phones. Publicly available datasets were used to validate the scheme. Authors have compared the working of some features on various machine learning algorithms. Ma et al. [13] have developed a message topic model (MTM) for detecting SMS Spam messages. MTM depends on the likelihood hypothesis of latent semantic analysis, symbol terms, background terms and topic terms to indicate spam messages. They have utilized the k-means algorithm to eliminate the sparse problem by grouping training spam messages into rough classes.

Hidalgo et al. [14] evaluated a few classifiers based on Bayesian learning in order to detect spam messages. The authors proposed two SMS spam datasets: the Spanish (199 spams and 1157 ham) and English (82 spam and 1119 ham) datasets. Feng et al. [15] have developed a Support Vector Machine based Naïve Bayes Algorithm (SVM-NB) for filtering spam emails. This algorithm is only applicable to text-based spam email detection, and it initially classifies training samples using the original Naïve Bayes algorithm and then constructing a hyper lane divides the training set into 2 parts i.e. spam and ham. They have utilized trimming strategy to reduce the size by eliminating ruthless samples from the training data.

3 Proposed Spam Detection Approach

The framework presented in this paper identifies a set of features that classify the messages as ham or spam accurately. Figure 3 shows the architecture of the proposed scheme. The feature selection is an important part and each feature selected is scrutinized properly before considering them. In addition, the messages are normalized before selecting a feature from them. Hence, the normalized data is merely a set of important words extracted from the entire message. This section discusses the normalization process, various features selected for detecting spam messages, Machine learning algorithm used and the proposed application.

3.1 Text Normalization

Text normalization is the process of converting the message into a more understandable form. Unstructured and unconstructed language, homonyms, abbreviations, and other casual messaging methods are commonly used by users in text based communication due to which it becomes extensively difficult to analyze the message effectively. In order to make the process of spam detection effective, a number of text normalization steps

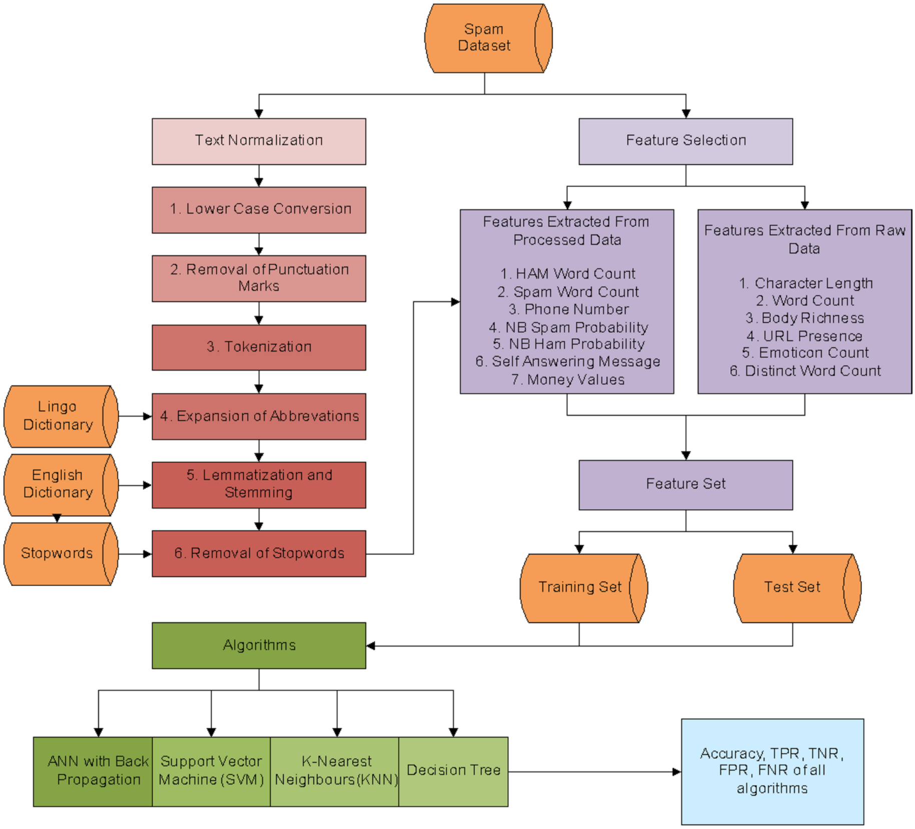


Fig. 3 The architecture of the proposed scheme

are performed which are discussed in this section. For better understanding of the normalization process, an example is shown in Fig. 4. Following steps are involved in the normalization process.

Conversion to Lowercase In this step, the entire message is converted to lowercase. The aim of this step is to avoid any demarcation between same words which differ in case only. For instance, words “Money” and “money” might be considered different from each other by the machine learning algorithms during the training period so this step will convert the message into a more consistent form.

Removal of Punctuation Mark In this step, the punctuation marks are removed from the message to convert into a set of words with no sentence separation. With this, the words followed by a punctuation mark are not considered as a different entry from the same word with no punctuation mark. For example, in the messages *1. “He plays*.” and *2. “He plays with a ball.”,* “plays.” in example (1) and “plays” in example (2) will be recognized separately if punctuation removal is not implemented.

Tokenization Once the punctuations are removed, the message is left in form of a set of words. Through tokenization, these words are assembled to form of an array of length equal to the number of words in the message. This step is important because it makes each word in the message easily accessible.

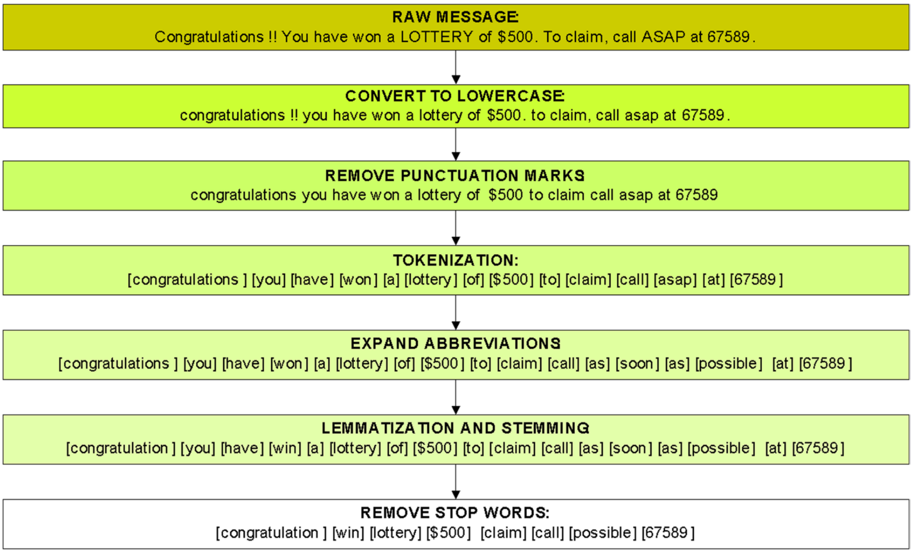

RAW MESSAGE:
Congratulations !! You have won a LOTTERY of $500. To claim, call ASAP at 67589.
CONVERT TO LOWERCASE:
congratulations !! you have won a lottery of $500. to claim, call asap at 67589.
REMOVE PUNCTUATION MARKS
congratulations you have won a lottery of $500 to claim call asap at 67589
TOKENIZATION:
[congratulations] [you] [have] [won] [a] [lottery] [of] [$500] [to] [claim] [call] [asap] [at] [67589]
EXPAND ABBREVIATIONS
[congratulations] [you] [have] [won] [a] [lottery] [of] [$500] [to] [claim] [call] [as] [soon] [as] [possible] [at] [67589]
LEMMATIZATION AND STEMMING
[congratulation] [you] [have] [win] [a] [lottery] [of] [$500] [to] [claim] [call] [as] [soon] [as] [possible] [at] [67589]
REMOVE STOP WORDS:
[congratulation] [win] [lottery] [$500] [claim] [call] [possible] [67589]

Fig. 4 Example of the message processed through text normalization

Expansion of abbreviations The array of words composed in the previous step, comprises of many abbreviations which are not recognized during analysis. Hence, it is necessary to expand these abbreviations into their full forms. For this purpose, two dictionaries are used. The English Dictionary [16] converts all the abbreviations recognized by the English language to their full forms and other is Lingo Dictionary [17] that converts the commonly used abbreviations into their expanded forms. For example, the word "lol" is converted into "l*augh out loud"*.

Lemmatization and stemming Lemmatization is the process of converting the words to their root form. An English dictionary is used for this purpose which has the mapping of each word to its root form. This is necessary because converting the words to root form will help in classifying a broad range of words into a single class. For instance, all the words like "4get", "forgave", "4g8", forgiven" will be converted into a single word "forgive".

Removal of stop words Stop words are the words, the presence of which does not help in categorizing the message into ham or spam. These are sentence constructing words and does not significantly contribute in the characteristics of a particular class of message. List of the stop words used in text normalization is shown in Table 1.

Table 1 Stop words used in the text normalization

i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, or, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now

3.2 Feature Selection

After analysing spam messages, we have come up with various feature that are able to detect spam messages. Out of various identified features, we found that 14 features are most effective in detecting spam messages so we incorporated these features in the analysis procedure. The more number of features we use while classifying the messages into ham or spam, the easier will it be for the machine learning algorithm to differentiate between the categories. These 14 features that are used in the proposed scheme, every one of them is able to correctly classify the majority of text messages on its own. These features are important for classifying the messages into ham and spam categories. Selected features are discussed below.

Feature 1: Number of Ham words in a message Words that people normally use while texting can be seen occurring in legitimate messages more often than they occur in spam messages. The list of these words is called Ham words. To categorize a message, it is important to analyze the occurrence of ham words in these messages. After carefully studying the test dataset, a list of Ham words is prepared and is used to check the count of Ham words in messages. More number of ham words indicate the legitimacy of SMS. For this purpose, the dataset is analyzed for such words and top 100 words with maximum count are chosen. These words tend to appear frequently in ham messages. For example words like “happy”, “love”, “sorry”, “tomorrow” is selected. The count of these words in a message is selected as a feature.

Feature 2: Number of Spam words in a message Similar to the Ham words, words that more frequently occur in fraudulent messages than in legitimate messages are called Spam Words. These words indicate that the SMS is fake and the sender wants to obtain personal information from the receiver. These messages generally include words like “claim, jackpot, money, bank, account” etc. Similar to the ham words list taken from the ham messages present in the test dataset, the spam messages from the test dataset is considered to produce the frequency of most occurring spam words in any message. From the dataset, the most frequent 100 spam keywords are selected whose occurrence in a message act as a feature during the analysis. The higher number of spam word count in an SMS increases its probability to be a Spam Message.

Feature 3: URL Presence Most often, Spammers include a URL link to a fake website in SMSs trying to persuade the receiver into clicking those links which might lead to a virus being downloaded in the system or a form asking for receiver’s personal information or anything else that may try to hinder the privacy of receiver. After analyzing the dataset it was observed that true test messages rarely included an URL while the spam messages usually include a URL asking receiver for an immediate call of action. Hence, the presence of URLs is considered an important feature while classifying the messages.

Feature 4: Presence of contact numbers Spammers often try to fool users by asking them to call instantly to claim a prize or lottery. Spam messages frequently include phrases like “Give a missed call on 999XXXXXXX to claim your prize!!!”. This is done to induce the user into giving their time and attention towards fake propositions and then afterward fooling or blackmailing the receiver into make them reveal their secret information. Hence, if contact numbers is present with other features like money values and URLs, then it can contribute as an important feature to detect spam messages.

Feature 5 and 6: Naïve Bayes Probability Naive Bayes classifier is based on Bayes theorem with strong independence assumptions among the features. It is a simple probabilistic classifier. The main advantage of the Bayes classifier is that it needs only a small amount of training data to estimate the parameters necessary for classification.

$$p(Tk|n) = \frac{p(m|Tk)p(Tk)}{p(m)} \tag{1}$$

Equation 1 gives the conditional probability of the message belonging to either ham or spam category. P(m) is the probability of the existence of message which is 1 here as the message is surely present. P(Tk) is the probability of the existence of a particular category in the entire dataset. P(m|Tk) is the probability of the presence of the message in a particular category. Naive Bayes Classifier is a simple and easy classifier to implement. In addition, it outperforms many classification algorithms. It is also fast and works well when the dimensions of input is high. Using a classification algorithm as a feature also gives an additional advantage of incorporating two different machine learning algorithms into one where Naive Bayes becomes the supporting algorithm. Using Naive Bayes algorithm the ham and spam probability is found and used as two separate features in the proposed scheme: NB Ham Probability and NB Spam Probability.

Feature 7: Number of Emoticons People while texting like to use emoticons to express the sentiment of the message. However, spam messages lack the use of emoticons because these messages generally lack sentiments and spammers have to convey a large amount of information in a limited 180 character message length. Hence, emoticons become a great differentiator for the proposed scheme. The dictionary of emoticons is used to determine number of such characters in every message. The significance of the number of emoticons is given to the observation of having more emoticons in the ham messages than the spam messages. Algorithm 1 explains how the count of emoticons is extracted from a message.

Algorithm 1: Emoticons Count in Message

Input: Text message
Output: Number of emotion symbol

```
Set E_count = 0
Set Emoji_list = [ 💻,☎,:D :P, :/, <3 ]
for word x=1 to n
    if word ∈ Emoji_list
        E_count = E_count +1
    end if
end for
return E_count
```

Feature 8: Total Character Length The limit of one SMSs is 180 characters. The fraudsters usually send long text messages to the users to lure the as they need to provide users with loads of information. Also, spam messages does not consist of abbreviated words, due to which they have longer character length than ham messages. Ham messages use informal words which often have short forms of words and lack the use of vowels. For example, messages like “Are you happy?” can be written as “R u hpy?”. The length of the former is 14 characters while the latter has only 7 characters which is exactly half. Therefore, there are some regular patterns in the number of characters in messages of different genres. The purpose of keeping this as a feature is to scrutinize the effect of message length on the probability of ham and spam.

Feature 9: Number of words The number of words actually determines how long the message is. Illegitimate message tends to be longer than the legitimate messages as they often try to provide maximum information to lure the receiver while the legitimate messages are shorter and abbreviated with minimal information. To keep this information intact with the analysis procedure, we have taken the number of words as a feature.

Feature 10: Body Richness Body richness is defined as the ratio of the number of characters in the message to the number of words present in it. A higher value of body richness indicates lesser words with long spelled words are present in the message. A small value of body richness indicates that a lot of short words are used. As the number of characters and number of words act as differentiators between the spam and ham messages similarly, body richness also indicates the same. A high value of body richness validates that more information is being transferred to the receiver, while a low body richness indicates lesser information is being transferred to the receiver. Spam messages tend to have the high amount of information. Hence, a higher value of body richness indicates that the message is spam else it is ham. This feature works upon the dataset to decide the fashion in which ham and spam messages are usually written. Algorithm 2 describes the procedure to calculate body richness. In algorithm 2, get_wordCount() return the count of words in a message.

Algorithm 2: Calculate Body Richness

Input: Text message
Output: Body Richness

```
char_length = msg.length
word_count = get_wordCount(msg)
body_richness = char_length / word_count
return body_richness
```

Feature 11: URL Content Type Only the mere presence of a URL link in the test message is not enough for classification purpose. It is important to check if the URL is malicious or not. The content of the URL must be checked to see if it returns a safe HTML page or leads to a download of malicious files. Some of these APK files are malicious in nature. On clicking some link, a download might also start in the background which could install viruses or malware that can steal user's bank account details, passwords, and other personal information. Hence whether the URL link in a text message leads to a download of some malicious file, should be considered as an important feature while classifying the messages into ham or spam.

Feature 12: Number of distinct words Spam messages often use more and distinct amount of words to convey their messages to the user. The number of distinct words is counted by considering each word only once within a message. A number of distinct words indicate more data in the message. Algorithm 3 describes the process of counting distinct words in a message. A dictionary named unique is build that contains only unique words and the repeating words are not added to it. Length of this dictionary indicates total unique words in a message.

Algorithm 3: Count Distinct Words

Input: Text message
Output: number of unique word in message

```
set unique[] = 0
for word x=1 to n
    if word ∈ unique:
        continue
    else
        unique ←unique U word
    end if
end for
Return unique.length
```

Feature 13: Presence of Money Values. Main aim of most of the spammers is to gain personal information by offering the receiver some instant financial benefits and for this purpose they use currency symbols and words like "cash", "money" in the messages. The presence of symbols like Rs*, $, £* and the word like *"cash"* is considered as a feature to mark messages which claim money from the recipients and are usually spam. Spammers often use these values for claiming some prize money or jackpot of worth millions to lure receivers. Algorithm 4 shows how the presence of money values is determined in a message. As soon as any money related symbol is found, the algorithm immediately returns 1 indicating that the message consist of money value and if the algorithm returns 0, it indicates that there is no such value.

Algorithm 4: Check for money values

Input: Text message
Output: existence of money token

```
Set money_val = ["$","£","¥","€","cash","Rs"]
for word x=1 to n
    if word ∈ money_val
        return 1
    else
        return 0
    end if
end for
```

Feature 14: Self-Answering Message Self-answering messages are those messages that include some kind of call of action, that generally want the receiver to interact with the sender of the message in some or the other way. For example, making a call, sending an immediate reply, or opening a URL link in the message. Ham messages are generally informative in nature. This information is helpful to differentiate between the spam and ham messages very accurately. Self - answering messages are the messages that lure users about some cash prize, call or text immediately, or for claiming some kind of prize. If the message is Self-answering messages then it is likely to be a spam message.

3.3 Training Dataset Creation

For feature selection, both the raw data and processed data are used to select the features from the messages. A total of 14 features are selected from single message and these features are then appended back to back in the array called F.vector. The feature vector of an individual message is then appended to the array Feature.set to make the complete two-dimensional feature set, where the number of rows is equal to the number of messages in the dataset and the number of columns is equal to 14 (number of features selected from a message). Some features that needs to be extracted from the original messages are altered during text normalization, therefore it is necessary to use both raw message and normalized message in feature selection. Feature number 1–7 are selected from raw data includes features like the original length of the message, word count, presence of URL etc. Feature numbers 8–14 are extracted from processed data and includes features like the number of ham words and spam words contained in a message.

Table 2 shows examples of messages one of which is a spam message and the other is ham message. These messages are first processed under the text normalization phase and the feature selection procedure is applied to these processed messages. The resultant feature vector of these messages is also shown in the table. The length of the feature vector is 14, where one numerical value corresponds to a feature selected from the message.

4 Training with Back Propagation Artificial Neural Network

The back-propagation technique of artificial neural networks uses three completely different layers. The first layer represents the input layer, the second layer is a hidden layer and the third layer is the output layer. Every node can be considered as a vegetative cell. The output layer has two neurons that represent the two values 0 for ham and 1 for spam.

The values in the input layer are increased with weights to provide the output of a neuron. The result is then normalized between zero and one by passing through the sigmoid operation. Similarly, weights are computed for the hidden layer. After that the error i.e. the distinction between the actual output and the expected output is calculated and the weights of the input layer and hidden layer are adjusted per the distinction [18]. This process is repeated until the error is decreased and then the weights are used to test the data. The process is shown in Fig. 5.

Table 2 Example of formation of the feature vector

Messages	Category	Feature vector
As I entered my cabin my PA said, ‘’ Happy B’day Boss !!’’. I felt special. She askd me 4 lunch. After lunch she invited me to her apartment. We went there.	Ham/0	[2, 0, 0, 0, 1.92155303845e-31, 6.42238604722e-31, 0, 72, 12, 6, 0.0, 12, 0, 0]
If you don’t, your prize will go to another customer. T&C at www.t-c.biz 18 + 150p/min Polo Ltd Suite 373 London W1 J 6HL Please call back if busy.	Spam/1	[4, 2, 1, 0, 5.91977282461e-39, 8.8153183934e-48, 0, 154, 28, 5, 0.0, 27,0, 1]

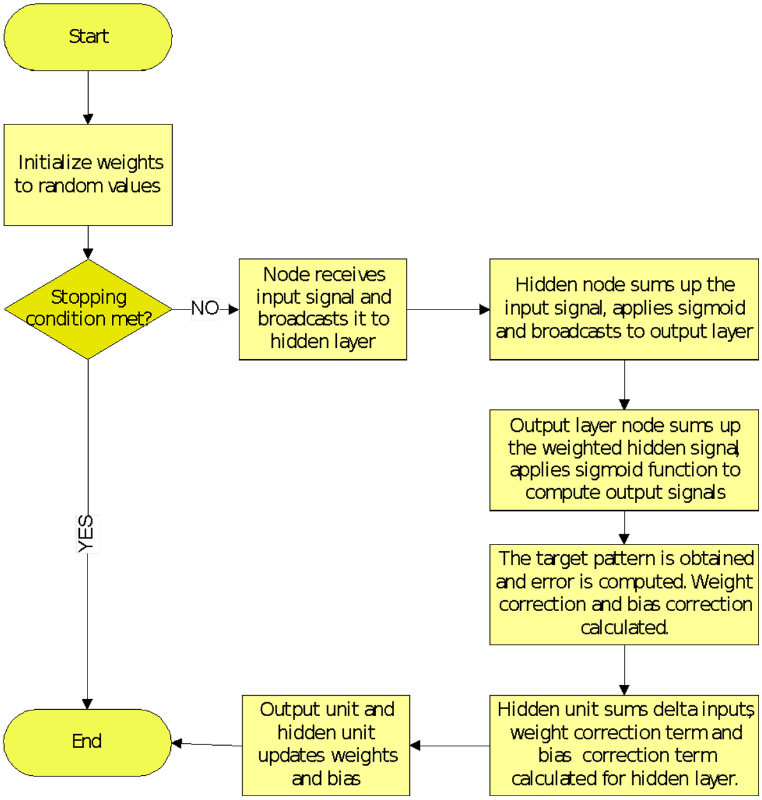

Start
Initialize weights to random values
Stopping condition met?
NO
YES
Node receives input signal and broadcasts it to hidden layer
Hidden node sums up the input signal, applies sigmoid and broadcasts to output layer
Output layer node sums up the weighted hidden signal, applies sigmoid function to compute output signals
The target pattern is obtained and error is computed. Weight correction and bias correction calculated.
Hidden unit sums delta inputs, weight correction term and bias correction term calculated for hidden layer.
Output unit and hidden unit updates weights and bias
End

Fig. 5 ANN algorithm using Back Propagation Technique

5 Performance Evaluation and Analysis

For experimental analysis and evaluation of the proposed framework, availability of dataset with an adequate number of instances plays an important role. In this work, SMS spam collection v.1 [19] is used as a dataset for experimental analysis. This is publically available dataset and consists of text messages from various sources [20–22] in the English language. It consists of 5574 text messages out of which 4827 are ham messages and 747 are spam messages. These datasets are combined together to make a larger dataset that contains a total out 10,728 messages. Out of these 10,728 messages, 1109 are spam messages and the rest 9618 messages are classified as ham messages. Weka [23] is a suite of machine learning algorithms written in Java. This is free software developed at the University Of Waikato, New Zealand. We have used this tool to determine the accuracy of algorithms stated over the feature set developed in this paper.

In the proposed approach, normalized data is analyzed over the identified features using various algorithms. To analyze the accuracy of the features, the following comparative studies have been made.

5.1 Pearson Product-Moment Correlation Coefficient (PCC)

It is the measure of the strength of linear association between two variables and is denoted by r. The PCC value (r) can range from + 1 to − 1. If the value is greater than 0, it indicates a positive association between two variables i.e. as the value of one variable increases, the value of other variables also increases. Similarly, value less than 0 indicates negative association which means that as the value of one variable increases, the value of second variable decreases. If the PCC value of 0, that means there is no association among the two variables. High values of PCC are always desired whether they are positive or negative because higher values indicate higher levels of association among variables.

Here, the values of the feature selected from the messages in the dataset and the categories to which they belong determined. It can be easily observed that feature number 2, 4, 13 and 14 show higher association with the dataset categories, this means that these features are of more importance than that of others and play a more significant role in determining the category of the text messages as shown in Fig. 6.

5.2 Feature Analysis

The problem statement stated in this paper requires the use of a supervised learning algorithm. Supervised learning algorithms are used in cases where you have an input variable **X** (in our case the feature vector for SMS) and an output variable **Y** (category of SMS i.e. Spam or Ham), and the goal is to find a mapping function **f** and approximate it so well such that **Y = f(x)** can be used to successfully predict the category of unknown SMSs. All the machine learning algorithms discussed below fall under the category of supervised learning algorithms and are considered to be the most efficient and powerful algorithms to solve the stated problem statement.

The KNN algorithm makes predictions for a new data point by searching through the entire training set for the K most similar instances and summarizing the output variable for those K instances.

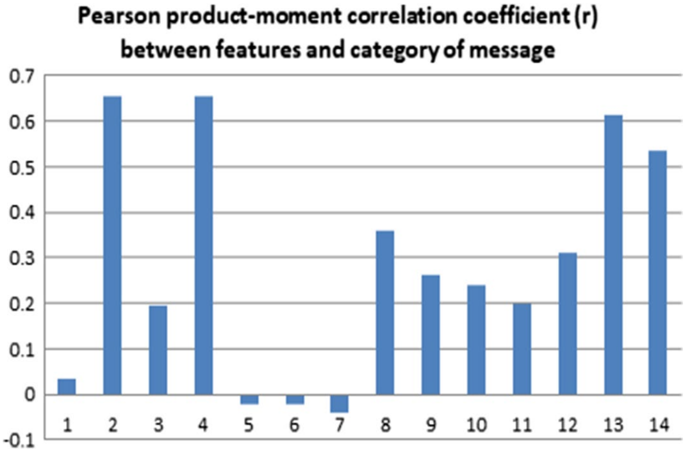


Fig. 6 PCC values of features

Decision Tree is also an important type of algorithm for predictive modelling machine learning. The representation of the decision tree model is a binary tree. Each internal node of the tree corresponds to an attribute associated with the training dataset, and every leaf node corresponds to a class label. In starting, the entire training set is taken into account as the root of the tree. Decision Tree Algorithm works on recursively by selecting the best attribute to divide the data and expanding the leaf nodes of the tree until the stopping criteria are met.

Support Vector Machine is one of the most popular machine learning algorithms. The training messages are plotted as points of the graph so that the messages in different categories can be divided by a clear gap that is as far as possible. The gap can be seen as the hyper plane that separates ham messages from spam messages. SVM uses kernel functions to map training vectors into higher dimensional space. There are different types of kernel functions available like linear, polynomial, Radial Based Function (RBF), and sigmoid which can be used according to the characteristics of the dataset. This analysis has used Linear Kernel Function in order to train the dataset [8].

Back propagation algorithm as described above in Sect. 4 works by repeating the computations over multiple layers until the desired result is achieved.

The graph given in Fig. 7 gives the comparative study of the different features on each algorithm. It can be seen that feature number 4 and 13 independently can identify an illegitimate message with nearly 94% accuracy. Feature 5 shows the minimum accuracy of 89.07% on the K-nearest neighbour algorithm, while on the rest of 3 algorithms the accuracy is 89.66%. Thus, all the features selected from the dataset are able to identify most of the messages individually indicating that these features are of great importance in classifying the messages.

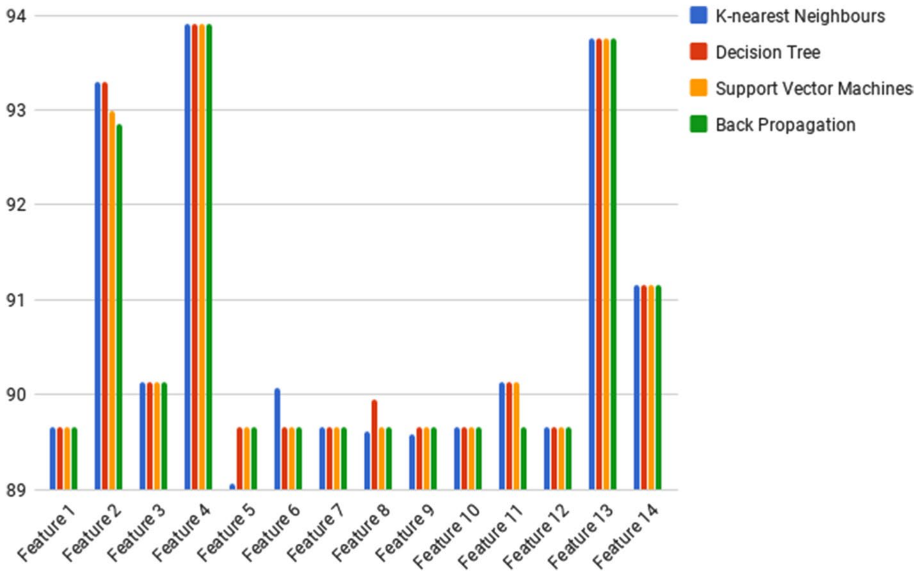


Fig. 7 Comparison of all features in different algorithms

Table 3 Implementation of feature selection on the machine learning algorithms

Algorithms	Accuracy (%)	True positive rate (%)	True negative rate (%)	False positive rate (%)	False negative rate (%)
Decision tree	95.76	97	85	15	3
Support vector machine	95.18	98	74	26	2
Back propagation	95.81	97	89	11	3
K-nearest neighbors	95.42	97	80	20	2

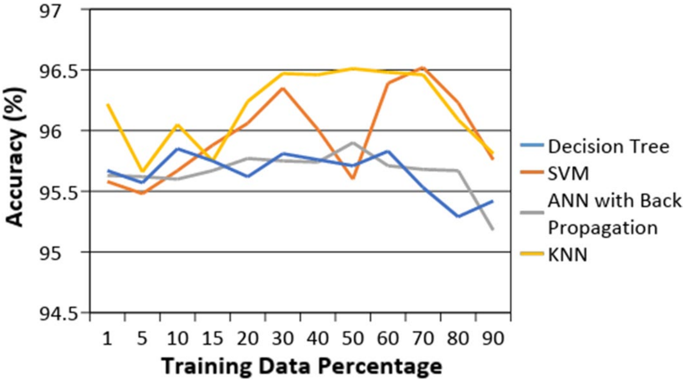


Fig. 8 All algorithms at different training percentages

5.3 Classification Results

The entire feature vector is applied upon the algorithms to give the result stated in Table 3. 90% of messages i.e. 9656 messages of the dataset is split into a training set and the rest i.e. 10% messages i.e. 1072 messages are used for testing purpose. It can be seen that the artificial neural network using back propagation technique performs the best with an accuracy of 95.81% and SVM is the worst with 95.18%. However, SVM still manages to maintain the highest True Positive Rate of 98%, while the rest are at 97%. The false negative rate is significantly low in every algorithm i.e. within the range of 2–3%.

5.4 The Accuracy of Algorithms on Different Training-Test Set Split Percentage

All the algorithms are tested on the dataset with different training and test set split percentages. Figure 8 shows the line graph of the same. It is clearly depicted in the figure that when only 1% of the dataset is used for training the algorithms (randomized selection), ANN with Back-propagation achieves maximum accuracy i.e. 96.22%. Decision Tree achieves the maximum accuracy of 96.52% when training data percentage is 70%. The accuracy of SVM remains constant and does not vary much with a change in the split percentage.

Table 4 Time elapsed by machine learning algorithm

Algorithms	Elapsed training time	Elapsed testing time	User CPU training time	User CPU testing time
Decision tree	0.51	0.00	509	1.00
Support vector machine	1.99	0.00	1984	4.00
Back propagation	6.97	0.00	6950	2.00
K-nearest neighbors	0.00	1.97	1.00	1764.00

Table 5 Accuracy comparison

Algorithms	Accuracy (%)
SNAP [24]	83.9
AirSenti [24]	80.5
Naïve Bayes [25]	64
Random Forests [25]	63
Proposed approach (ANN with BP)	95.81

5.5 Comparison of Time Elapsed

The time elapsed by different algorithms is given in Table 4. K-Nearest Neighbor's algorithm takes minimum training time because it does not have any training phase. However, the testing time in K-nearest neighbors is significantly larger than the other because the distance of a single message is measured from all 10,000 messages in the training set, to find the k-nearest neighbors. Maximum time in the training phase is taken by the back propagation algorithm as it improves by iterating upon the errors calculated during the training phase but the time taken is backed up by the maximum accuracy achieved by the algorithm.

Xu Chi et al. in [24] have used a two-stage adaptive feature selection algorithm in which they have assigned weights to the features depending on their importance. For the purpose of sentiment detection, we have implemented KNN Machine Learning Algorithm. The algorithm with a value of $k=20$ achieves the accuracy of 83% which is quite low as compared to the accuracy achieved by our scheme. Similarly, Etaiwi et al. [25] discussed the effects of feature selection on spam detection performance. They have used Bag of Words and Part of speech tagging as their features along with linguistic features and word counts. These features are tested upon four different algorithms namely Naïve Bayes, Decision Tree, Support Vector Machine and Random Forests which achieves an accuracy of 64%, 61%, 49%, and 63% respectively as shown Table 5. It lacks many important features like presence of URLs or money values, which clearly indicates that a message is illegitimate.

6 Conclusion

Number of Spam messages are increasing at an alarming rate due to the availability of SMS message packages at low price. SMS is one of the most trustworthy text-based communication channel. Also SMS messages have higher response rates as compared to other text based communication channels due to which it is highly preferred by the attackers to spread spam messages using text SMS. So, a technique is required that is able to detect spam messages with high accuracy. The model proposed in this paper proves to be an efficient method for detecting spam messages. The existing current applications available to detect spam messages works on the basis of user reviews. However, the algorithms given by us analyze the message on the basis of various features identified by us that are able to effectively detect spam messages. In addition, we have implemented and tested our scheme on four machine learning algorithms and results shows that all the features are not equally effective. There are some features like the presence of phone numbers and money values that contribute a more to the result and there are some features that have a very less influence.

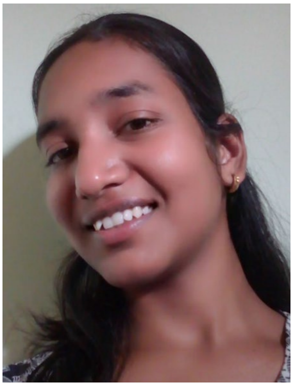
The proposed application can also be turned dynamic to improvise the feature vector after adding testing messages into the database. Once a particular number of messages are stored in the database, the feature vector can be rebuilt. The paper deals with different features and it is clearly visible that all the four algorithms used, work almost the same with an approximate accuracy of 95% but back propagation algorithm gives the highest accuracy i.e. 95.81% with 97% True positive rate and 89% True negative rate. In future, considering the impact of Smishing attack in the today's world, we will work on introducing new features that will be able to detect smishing messages. Although smishing message is the subset of spam message but due to increasing popularity of text messages, attackers are carrying out phishing attack via SMS. So, this attack needs to be addressed. So, we will extend our scheme so as to detect smishing messages along with the spam messages.

References

1. Goel, D., & Jain, A. K. (2018). Mobile phishing attacks and defence mechanisms: state of art and open research challenges. *Computers & Security, 73,* 519–544.
2. Gudkova, D. (2018). *Spam and Phishing in 2017*. Securelist—Kaspersky Lab's Cyber threat Research and Reports, 15 Feb, 2018, http://securelist.com/spam-and-phishing-in-2017/83833/. Retrived April, 2018.
3. Crowe, J. (2017). Must-Know Phishing Statistics 2017. Barkly Endpoint Security Blog. http://blog.barkly.com/phishing-statistics-2017. Retrived April 2018.
4. Goel, D., & Jain, A. K. (2017). Smishing-classifier: A novel framework for detection of Smishing attack in mobile environment. In *International conference on next generation computing technologies* (pp. 502–512). Singapore: Springer.
5. Almeida, T. A., Silva, T. P., Santos, I., & Hidalgo, J. M. G. (2016). Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. *Knowledge-Based Systems, 108,* 25–32.
6. Joo, J. W., Moon, S. Y., Singh, S., & Park, J. H. (2017). S-detector: An enhanced security model for detecting Smishing attack for mobile computing. *Telecommunication Systems, 66*(1), 29–38. https://doi.org/10.1007/s11235-016-0269-9.
7. Etaiwi, W., & Awajan, A. (2017). The effects of features selection methods on spam review detection performance. *International Conference on New Trends in Computing Sciences (ICTCS)*. https://doi.org/10.1109/ictcs.2017.50.
8. Patel, R., & Thakkar, P. (2014). Opinion spam detection using feature selection. *International Conference on Computational Intelligence and Communication Networks.*. https://doi.org/10.1109/cicn.2014.127.
9. Ali, S. S., & Maqsood, J. (2018). Net library for SMS spam detection using machine learning: A cross platform solution. *15th International Bhurban Conference on Applied Sciences and Technology (IBCAST).*. https://doi.org/10.1109/ibcast.2018.8312266.
10. Jain, A. K., & Gupta, B. (2018). Rule-based framework for detection of Smishing messages in mobile environment. *Procedia Computer Science, 125,* 617–623. https://doi.org/10.1016/j.procs.2017.12.079.
11. Wu, T., Liu, S., Zhang, J., & Xiang, Y. (2017). Twitter spam detection based on deep learning. *Proceedings of the Australasian Computer Science Week Multiconference on—ACSW*. https://doi.org/10.1145/3014812.3014815.
12. Sethi, P., Bhandari, V., & Kohli, B. (2017). SMS spam detection and comparison of various machine learning algorithms. *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 1,* 1. https://doi.org/10.1109/ic3tsn.2017.8284445.

13. Ma, J., Zhang, Y., Liu, J., Yu, K., & Wang, X. (2016) Intelligent SMS spam filtering using topic model. In *International conference on intelligent networking and collaborative systems (INCoS)* (pp 380–383). IEEE.
14. Gómez Hidalgo, J. M., Bringas, G. C., Sánz, E. P., & García, F. C. (2006). Content based SMS spam filtering. In *ACM symposium on Document engineering* (pp. 107–114). ACM.
15. Feng, W., Sun, J., Zhang, L., Cao, C., & Yang, Q. (2016). A support vector machine based naive Bayes algorithm for spam filtering. In *35th International conference on performance computing and communications conference (IPCCC)* (pp. 1–8). IEEE.
16. FreeLing. (2018). http://devel.cpl.upc.edu/freeling/. Retrived April 2018.
17. Internet & Text Slang Dictionary, www.noslang.com/dictionary/. Retrived April, 2018.
18. Gupta, S., & Singhal, A. (2017). Phishing URL detection by using artificial neural network with PSO. In *2nd International Conference on Telecommunication and Networks (TEL-NET)* (pp. 1–6). IEEE.
19. SMS Spam Collection. http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. Retrived April, 2018.
20. Grumbletext UK Forum. http://www.grumbletext.co.uk/. Retrived April, 2018.
21. A corpus linguistic study of SMS Text Messaging. http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf. Retrived April, 2018.
22. NUS Natural Language Processing Group. http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/. Retrived April, 2018.
23. Frank, E., Hall, M. A., & Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"* (4th ed.). Burlington: Morgan Kaufmann.
24. Chi, X., Siew, T. P., & Cambria, E. (2017). Adaptive two-stage feature selection for sentiment classification. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. https://doi.org/10.1109/smc.2017.8122782.
25. Etaiwi, W., & Awajan, A. (2017). The effects of features selection methods on spam review detection performance. *International Conference on New Trends in Computing Sciences (ICTCS)., 1,* 1. https://doi.org/10.1109/ictcs.2017.50.

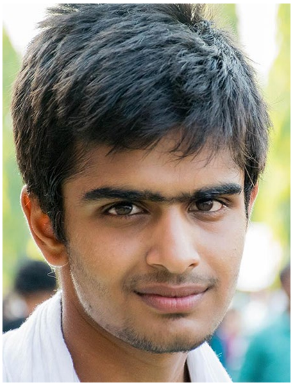
Ankit Kumar Jain is presently working as Assistant Professor in National Institute of Technology, Kurukshetra, India. He received Master of technology from Indian Institute of Information Technology Allahabad (IIIT) India and Ph.D. degree from National Institute of Technology, Kurukshetra. His general research interest is in the area of Information and Cyber security, Phishing Website Detection, Web security, Mobile Security, Online Social Network and Machine Learning. He has published many papers in reputed journals and conferences.



Diksha Goel has received her M.Tech. degree in Computer Engineering (Specialization in Cyber Security) from National Institute of Technology, Kurukshetra, Haryana, India. Currenly, she is pursuing PhD from the University of Adelaide, Australia. Her research interest includes Mobile security, Web security, Machine learning and Smishing detection.

Sanjli Agarwal completed Bachlor of Technology (B.Tech) in Information Technology from National Institute of Technology, Kurukshetra in 2018. Her research includes mobile security, spam detection and machine learning.

Yukta Singh completed Bachlor of Technology (B.Tech) in Information Technology from National Institute of Technology, Kurukshetra in 2018. Her research includes mobile security, spam detection and machine learning.



Gaurav Bajaj completed Bachlor of Technology (B.Tech) in Information Technology from National Institute of Technology, Kurukshetra in 2018. His research includes spam and phishing detection, machine learning and smartphone security.