March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

Rumour Detection on Twitter Using Long Short-Term Memory

Aparna Bindage
B.E in Information Technology Sinhgad College Of Engineering Pune, Maharashtra, India aparnabindage@gmail.com
S. M. Jaybhaye
Assistant Professorin Information Technology Sinhgad College Of Engineering Pune, Maharashtra, India
Sai Pande
B.E in Information Technology Sinhgad College Of Engineering Pune, Maharashtra, India saipande13@gmail.com
Datta Dhebe
B.E in Information Technology Sinhgad College Of Engineering Pune, Maharashtra, India
dattadhebe75@gmail.com

Article Info
Volume 83
Page Number: 2131 - 2138
Publication Issue:
March - April 2020

Article History
Article Received: 24 July 2019
Revised: 12 September 2019
Accepted: 15 February 2020
Publication: 18 March 2020

Abstract
Microblogging platforms are perfect place for breeding and spreading rumours. Propagation of these rumours at an alarming rate has become a severe social issue which has an adverse effect on people and organizations. Therefore, it becomes important to detect rumours quickly, automatically and increase the accuracy of learning model than the existing models which uses feature engineering that is biased, labour intensive and time-consuming. Even though, recent studies uses machine learning-based methods for automatic rumour detection by extracting features of rumour contents (e.g., people's opinions, questions, etc.) and static spreading processes, early detection of rumour remains a challenge. This paper proposes a learning model, Long Short-Term Memory (LSTM) combined with pooling operation of Convolutional Neural Network (CNN) for early detection of rumour. LSTM networks are well-suited for processing, classifying and making predictions based on time series data and also deals with vanishing and exploding gradient problems that can be encountered when training traditional Recurrent Neural Network (RNN). The dynamic changes of forwarding contents and spreaders are taken into consideration using LSTM based model.

Keywords *– Rumour identification, Twitter, Long Short-Term Memory, Microblogging sites, Early detection*

I. INTRODUCTION

Nowadays, social media has flourished, especially microblogging sites such as Sina Weibo and Twitter have become popular platforms that facilitates fast dissemination and acquisition of information. Internet is easily accessible and so is social media. Today, Twitter has emerged as one of the prime social networking sites which is massively used for sharing tweets. These tweets may contain user's opinions, article links, photos, quotes and much more. Twitter is used extensively as a microblogging service worldwide due to its 280-character messages called Tweets. These microblogging sites are prone to rapidly spreading rumours which can cause harm to society.

A rumour is a statement which is circulated without confirming the facts [1]. Rumours arise in the context of ambiguity or when the situation is not clear to people [2]. Rumours hence are harmful force that affects people and organizations [3]. For example, a rumoured tweet saying two explosions in White House and Barack Obama is injured was released from the official Twitter account of the Associated Press (AP) which was hacked on April 23, 2013. This rumour

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

was reposted extensively and caused social panic immediately. The impact of this rumour was huge - S&P 500 Index immediately dropped by 14 points, demolishing $136.5 billion in just a few seconds.Finding and debunking rumours at a primitive stage of spreading should be given the highest priority for reducing their negative influence on society. Some organizations have arranged rumour querying and rumour denial websites like factcheck.org, snopes.com, so that people can confirm if the tweet is a rumour or a non-rumour. These websites generally rely on the manual verification and public report to find rumours, which is a labour-intensive procedure and requires a lot of funding. Besides these disadvantages, verifying rumours is time-consuming. Rumour detection model is built using LSTM networks combined with pooling function of CNN. This model relies on the contents of tweets and the spreader's information for debunking rumours. The model outputs whether the original tweet is rumour or non-rumour.

The work presented in this paper is summarized as follows:

- LSTM-based model performs significantly well than feature-based machine learning methods. Hence, the hidden characteristics and local information can be learnt by proposed model very well in the spreading of rumours and non-rumours.
- Further, this model enables early identification of rumours, which serves efficient as compared to existing methods of rumour detection services.
- In the experiments, the twitter dataset is adopted which is presented in [4] that contains 498 rumours and 494 non-rumours[1].

II. LITERATURE SURVEY

Ma et al. [4] first applied RNN to detect rumours. It was observed that a rumoured event consists of an original message and other messages like comments and reposts related to the event, which created a continuous stream of messages. Thus, they model the rumoured data in time series with variable length. In a time-series, a rumoured event consists of several messages, so they batch these messages into time intervals and consider them as one unit. Using three recurrent units via tanh, GRU and LSTM, RNN-based method is evaluated. GRU and LSTM units have the ability to significantly capture the long-term dependencies for the messages. In addition to this, the performance for detection of rumour at a primitive stage is also high. However, as time changes, users may post messages differently i.e. from expressing shock and surprise to questioning the credibility of the message. Hence, textual features from forwarding contents may change their value over time. Therefore, it is required to find which of those features are more important for the detection task. As social media contains massive unlabelled data the future work of this paper is to develop unsupervised model for identification of rumours.

Chen et al. [5] proposed a deep-attention model for early detection of rumours which is based on RNN. A deep attention-based RNN for early identification of rumours, namely CallAtRumours(Call Attention to Rumours) is proposed. First, the stream of messages is modelled into a continuous time series with variable length. Soft-attention mechanism is combined into recurrence to fetch out distinct features and avoid duplicity. The model identifies rumours by learning temporary hidden representations from the posts. In each time step, the hidden state will be allocated a weight parameter to calculate its significance and benefaction to the results. The words that express user's opinions like anger or doubt are given more weight while, unrelated words are ignored. The future work is to examine the possibility to unite

[1]http://alt.qcri.org/~wgao/data/rumdect.zip

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

complex features with the model as well as to analyse issue of efficiency by applying hashing techniques on multiple layers of features.

Weiling Chen, Chiew Tong Lau, Chai Kiat Yeo, Bu Sung Lee, Yan Zhang [6] proposed a rumour detection model using a combination of Autoencoder and RNN. Variations in user's behaviour are observed while commenting on rumour posts and sincere posts and are described using comment based features. Using RNN, analysis of the features is done by taking into consideration the features that vary over time. Methods using RNN finds rumour more rapidly and correctly than techniques which are already in existence. Time-dependent characteristics are merged with the time-independent characteristics which are extracted out of the original post and further fed as an input to an Autoencoder (AE) for detecting rumour. The proposed unsupervised model which combines RNN with Autoencoder helps to increase accuracy. This model fails in considering the dynamic variations among the spreaders and the diffusion structure.

Jing Ma, Wei Gao, Kam-Fai Wong [7] proposed a neural rumour detection model RvNN (a type of tree-structured neural network) that bridges propagation clues and semantics of content. This approach helps to learn distinct features from tweets by following their propagation structure which is non-sequential in nature and then generating stronger representations to classify various classes of rumours. Two models - bottom-up RvNN and top-down RvNN are proposed for identifying rumour and this work produces a more desirable integrated representations for a message by capturing structural and textural properties indicating rumors. Desirable performance is achieved on comparing with the current approaches. The rumours are detected in early stage. This model fails to observe dynamic differences between spreaders and diffusion structure. In future work, more characteristics like user properties, etc. will be integrated with structured neural models to improve accuracy.

Oluwaseun Ajao, Shahrzad Zargari, Deepayan Bhowmik [8] provided a framework that detects fake news feeds and classifies them using a hybrid of CNN and LSTM models on microblogging site like Twitter. Further, a max pooling layer is used to avoid over-fitting and reduce dimensionality and cost of detection. This learning approach achieved 82% accuracy on PHEME dataset. This approach first recognizes closely related features connected with fake messages without prior insight of the field and then detects and classifies rumours using both texts and images. The performance of this model was hindered up to some extent due to an insufficient amount of training dataset. The future work is to find out the origin of fake posts and its location.

III. BLOCK DIAGRAM OF SYSTEM

A rumour is an unverified statement which not necessarily be misinformation. On investigating or debunking a rumour, it can turn into a genuine fact or actual rumour (i.e. misinformation). This model focuses on detecting actual rumours since most of them have negative effects on society. In the later sections, the actual rumours are referred to as "rumour" and the rumours which turn out to be genuine are referred to as "non-rumours".

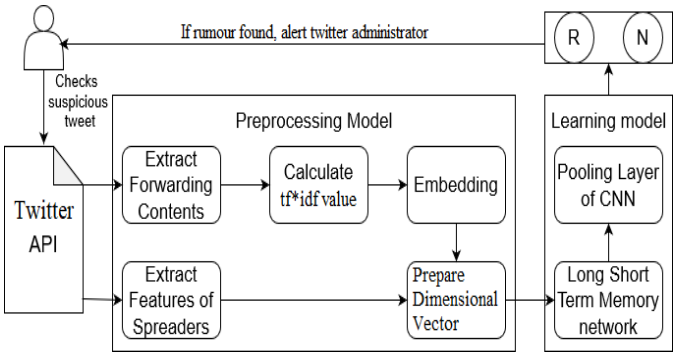

Fig. 1: Block diagram of rumour detection system

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

Fig. 1 depicts the block diagram of proposed rumour detection model. For a suspicious tweet submitted to system, proposed model examines for forwarding contents and spreaders information.

Phase 1: The content of tweets varies dynamically over period of time (e.g. doubting words in tweet). So, these strong signals need to be captured from forwarding contents. In the pre-processing phase, important features of forwarding contents are extracted by first calculating the *tf*idf* value and then using a word embedding that maps word to vector of numerical values. Embedding is essential because the input fed to LSTM should be numeric. Word2Vec model is used which is a shallow two-layered neural network model used to produce word embedding to represent words as numeric values. The *tf*idf* value and vectors of vocabulary is converted to low-dimensional representation and resulting dimension D_c is given as input to model.In the similar manner, important features of spreaders (e.g. popularity, follower quantity, description, location, type and activity) are extracted. A dimensional vector is prepared for the features of spreaders i.e. D_s. This vector is then fed to LSTM.

Phase 2: The learning phase of model has LSTM and pooling layer of CNN. The important characteristics of rumours can be captured using RNN that differentiate rumours from non-rumours. The traditional recurrent unit suffers from short-term memory. LSTM overcomes this drawback as it maintains a memory cell c_t at time t and has internal mechanisms like gates to regulate the flow of information. The output h_t of an LSTM unit is computed by the following equations [5].

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + V_i c_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma(U_f h_{t-1} + W_f x_t + V_f c_{t-1} + b_f), \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(U_c h_{t-1} + W_c x_t + b_c), \quad (3)$$

$$o_t = \sigma(U_o h_{t-1} + W_o x_t + V_o c_t + b_o), \quad (4)$$

$$h_t = o_t \tanh(c_t), \quad (5)$$

Where tanh and σ (logistic sigmoid) are activation functions used to squash values to help regulate the values flowing through the network. The forget gate f_t identifies the information that is not required and will be removed from cell state. The input gate i_t decides, what new information to store in the cell state. The next step is to update memory cell c_t by forgetting part of the existing memory and adding new memory $\bar{c}_t$. Finally, the output gate will only output the parts which are required. Therefore, the LSTM-based model is developed using features extracted from spreaders and forwarding contents.

For the tweets that have several reposts, some implicit hints can be seen at any instant of time during the process of spreading. To downsample an input representation and to better capture these hidden clues, adding pooling function of CNN helps to acquire influential local information. Finally, the model gives the output as "rumour" or "non-rumour".

IV. IMPLEMENTATION OF LSTM BASED MODEL

A. Problem Definition

The comments and spreaders information contain hints that are implicit which can be used to differentiate rumours from non-rumours. Each incident has several tweets that are related to it. The collection of these incidents are defined as $O = \{O_i\}$, here each incident $O_i = \{(p_{i,j}, s_{i,j}, t_{i,j})\}$ contains all relevant tweets $p_{i,j}$, spreaders $s_{i,j}$ at time $t_{i,j}$, where j=0 signifies that spreader $s_{i,0}$ posts the original tweet $p_{i,0}$ at time $t_{i,0}$. The aim is to determine whether each incident is rumour or not.

Input : Tweets of incident $O_i = \{(p_{i,j}, s_{i,j}, t_{i,j})\}_{j=1}^{m_i}$,
Reference length of LSTM L
Output: Time steps $S = \{S_1, S_2, \ldots\}$

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

$P(i) = t_{i,m_i} - t_{i,1}$; $p = \frac{P(i)}{L}$; $r = 0$;
while *true* **do**
r ++;
U_r ←*Equipartition (P(i), p)*;
U_0 ←{empty intervals}⊆ U_r ;
$U'_r \leftarrow U_r - U_0$;
Find $\bar{U}_r \subseteq U'_r$ so that $\bar{U}_r$ consists of continuous timesteps which cover the largest time-span;
if $|\bar{U}_r| < L$ && $|\bar{U}_r| > |\bar{U}_{r-1}|$ **then**
/* Halve the timestep */
$p = \frac{p}{2}$;
else
/* Give output */
$S = \{S_o \subseteq \bar{U}_r \mid S_1, \ldots, S_{|\bar{U}_r|}\}$;
return S;
end
end
return S;

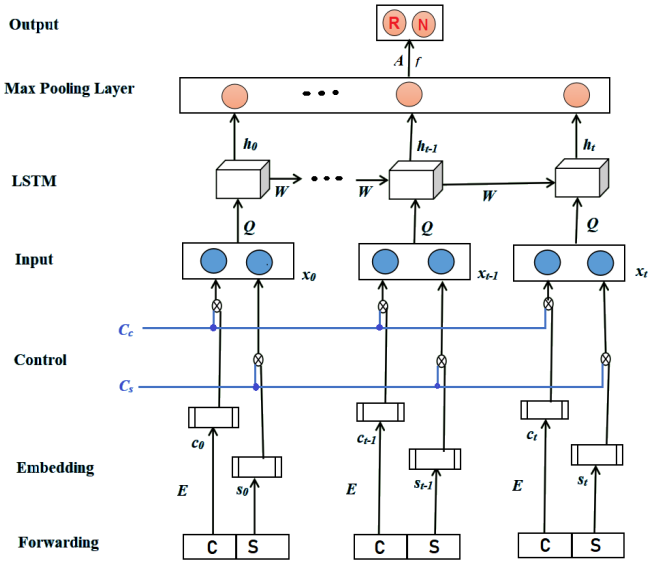

Fig.2: The LSTM-based rumour detection model

B. Variable Length Time-Series

Each tweet is modelled as an input instance and total number of tweets is considered as sequence length of LSTM. There are several tweets related to popular incident and there is only single output neuron representing the class for each incident. So, it is computationally expensive and also ineffective to carry out backpropagation through several timesteps with the final-stage loss. Hence, tweets are batched into time steps and then modelled using the LSTM. Time-span that corresponds to heavily populated tweets in the spreading need to be captured accurately. Algorithm I describes the method to construct variable length time series. Initially, divide the total timeline equally into L timesteps (i.e., L is the reference length). By removing the empty timesteps (i.e., each interval not having a single tweet) from the set U_0, system gets set of non-empty timesteps U'. Then, from set U' continuous timesteps are selected whose total time-span is the largest and stored into the set $\bar{U}$. If the timesteps in set $\bar{U}$ is lesser in number than L and more than that of previous round, timesteps are halved and are continued to partition; or else, the continuous timesteps is returned that is given by set $\bar{U}$.
construct time series of variable length.

C. Model Structure

The recurrent units of LSTM fit the timesteps naturally depending on time series constructed using Algorithm I. In each timestep, the dimensional vectors D_c and D_s are constructed. The random combination input of spreaders and contents of tweets are controlled using control layer that is added to the input of the proposed model.

$$x_t = c_t \cdot c_c \cup s_t \cdot c_s \qquad (6)$$

Equation (6) represents input x_t that is given to LSTM which keeps only relevant information to make predictions and forget non-relevant data. Here, c_s and c_c control the input of spreaders s_t and contents of tweet c_t respectively.For the tweets that have several reposts, some implicit hints can be seen at any instant of time during the process of spreading. To accurately get these implicit hints, pooling method of CNN is used. So, max pooling is used to choose special local features, which can further improve the performance of the rumour identification.

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

$$f_m = max\{h_{i,m}\}, \quad i \in 0...t \qquad (7)$$

Max pooling chooses a value for all dimension that is maximum of h_is which is represented by (7), where $i = 0...t$, m is the m^{th} dimension of h_i, f_m is value of max pooling for the m^{th} dimension. As, the output of proposed model is whether the tweet is rumour or not, softmax layer is used as activation function to obtain the probabilistic output of two classes.

$$V = Af + b_V, \qquad (8)$$

The input to the softmax layer is given by (8), where f is the value obtained from max pooling operation, A is the weight matrix, b_V is bias and V is the output of hidden layer, which is used to find whether the tweet is rumour or not.

$$p_i = \frac{e^{V_i}}{\sum_{k=1}^{n} e^{V_k}}, \qquad (9)$$

Equation (9) represents probability function of class i, where n is total number of classes and p_i is the probability associated with class i. The output, [1,0] is given by model that is used to represent the rumour class and [0,1] is used to represent the non-rumour class.

D. Model Training

The parameters of proposed model are trained using backpropagation and stochastic gradient descent (SGD) with mini-batch. Ada Grad algorithm is used for parameter update in training. The error between the value that is predicted and the value that is actual needs to be minimized. So, the loss function used here is cross entropy that is given by (10).

$$L(p,q) = -\sum_{i=1}^{c} p_i log\, q_i + \lambda \|\theta\|_2^2, \qquad (10)$$

where p is the value that is predicted, q is the value that is actual, θ and λ correspond to the model parameters and regularization coefficient respectively. Finally, model is iterated over all the training incidents in each epoch and execution is continued until +maximum epoch number is met or the loss value converges.

V. Result And Analysis

The hardware needed to run the experiment is 1.8GHz processors and 8 GB of RAM. LSTM model is implemented using python on Anaconda platform. Keras and TensorFlow are used for numeric computation and Google Word2Vec model is used for word embedding. Twitter API is used to access details of tweet. The twitter dataset presented in [4] is used in experiment which contains 992 events in total. Each line contains one event with the ids of relevant tweets i.e. event_id, label, tweet_ids. For the labels, the value is 1 if the event is a rumour, and is 0 otherwise. The tweet is downloaded using Twitter API. To provide an example we process on single event i.e. “PS 169 Principal Eujin Jaela Kim banned the Pledge of Allegiance, Santa, and Thanksgiving”. Tweets of user contains emojis, links, digits, special characters etc. which needs to be removed in pre-processing stage. The output obtained after pre-processing is shown in Fig. 3.

Fig.3: Removal of patterns, emojis and links

For this event several related tweet_ids are stored. For each tweet in tweet_ids list, the time at which it was posted is obtained. Then the corresponding tweet and time is sorted with respect to time, as we need to batch posts into time intervals according to algorithm I. The max continuous interval obtained is 4, so the output contains 4 documents.

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

[-0.05883789 -0.05786133 -0.06396484 0.32421875 0.11474609 -0.25195312 -0.09619141
0.22851562 -0.14550781 0.19726562 0.07617188 -0.21875 -0.45117188 0.00075912 0.10595703
0.36914062 0.03442383 -0.17285156. . . . . 0.14355469 -0.25195312 0.13964844 0.1796875]

Fig.5: Vector representation of word

In each interval, the *tf*idf* value of the vocabulary terms is calculated, then the vocabulary is pruned by keeping the top-K (here K = 500) terms according to their *tf*idf* values. This value signifies how important a word is to the document. Stop words are removed which take up space and valuable time of processor. Therefore, only important words are extracted by calculating *tf*idf* value. Fig. 4 shows unique words present in the document and *tf*idf* values are calculated for first document in descending order.

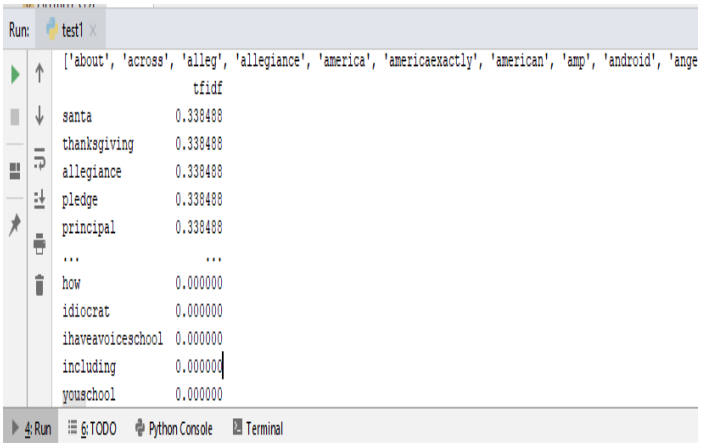

Fig.4: *Tf*idf* values in descending order

After getting important words, the words with similar meaning are depicted using similar representations using embedding. Embedding layer is included that converts the sparse input word vectors into low-dimensional representations. Embedding is done using Google Word2Vec model which provides an implementation of skip-gram and CBOW architectures. The vector representation of word "santa" with 300 dimension is shown in Fig.5.

Wordvector matrix is obtained as a result of embedding. Then data is divided into training and testing set in the ratio 8:2. For contents of tweets the training accuracy obtained is80%.

Following are the main features included in proposed model to improve performance of rumour detection than existing methods –

1. Considering only forwarding contents as input to model does not obtain unique effect on identification of rumours. So, it is important to take into consideration the features of spreaders for early identification of fake news and improve accuracy.

2. Early stopping is used to find the time required for training termination based on the effect of verification set, since deep learning tend to overfit easily on small datasets.

3. However, as time changes, users have tendency to post differently i.e. from expressing surprise to questioning. Hence, textual features from forwarding contents may change their value over time. So dynamic differences between spreaders is also taken into consideration.

4. Furthermore, addition of max pooling layer helps to obtain hidden clues in spreading process thereby increasing the performance of model and also reduce cost of model.

VI. CONCLUSION

The recent work on rumour identification from social media are based on machine learning-based methods that uses feature extraction which is biased, time-consuming and labour intensive. In

March-April 2020
ISSN: 0193-4120 Page No. 2131 - 2138

this work, a deep learning framework using LSTM and CNN for rumour debunking at an early stage is proposed. The dynamic differences in forwarding contents and spreaders are taken into consideration for precisely detecting rumours. Some spreaders play an important role in the process of spreading. The characteristics of spreaders influence significantly in differentiating rumours and non-rumours. Max pooling is used to reduce dimensionality and cost of the model as well as to improve the performance of rumour identification. Multiple hidden layers can be added to LSTM and also embedding layers can be added to improve the efficiency of model. The future work is to add more influential features to the model and also to take into consideration the correlated pictures associated with the tweet that will improve efficiency.

VII. REFERENCES

[1] G. Allport and L. Postman, "The psychology of rumour", Rinehart & Winston, 1947.

[2] N. DiFonzo, R. Rosnow and P. Bordia, "Reining in rumours", Organizational Dynamics, 1994, 23(1):47–62p.

[3] N. DiFonzo and P. Bordia, "Rumour psychology: Social and organizational approaches", American Psychological Association, 2007.

[4] J. Ma, B. J. Jansen, W. Gao, M. Cha, P. Mitra, K. Wong and S. Kwon, " Detecting rumors from microblogs with recurrent neural networks", in *Proceedings of IJCAI*, 2016.

[5] T. Chen, L. Wu, X. Li, Jun Zhang, Hongzhi Yin and Yang Wang, "Call attention to rumours: Deep attention based recurrent neural networks for early rumour detection", 2018, 40-52p.

[6] W. Chen, Y. Zhang, C. Yeo, C. Lau, B. Sung Lee. "Unsupervised rumour detection based on user's behaviours using neural networks", Journal paper *vol.105*, April 2018, 226-233p.

[7] J. Ma, K. Wong and W. Gao "Rumor Detection on Twitter with Tree-structured Recursive Neural Network'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, vol. 1*, 2018.

[8] O. Ajao, D. Bhowmik, and S. Zargari, "Fake news identification on Twitter with hybrid CNN and RNN models", in *Proc. 9th Int. Conf. Social MediaSoc.*, 2018, 226-230p.

1st Aparna Bindage
B.E in Information Technology
Sinhgad College Of Engineering
Pune, Maharashtra, India
aparnabindage@gmail.com

2ndSai Pande
B.E in Information Technology
Sinhgad College Of Engineering
Pune, Maharashtra, India
saipande13@gmail.com

3rdDatta Dhebe
B.E in Information Technology
Sinhgad College Of Engineering
Pune, Maharashtra, India
dattadhebe75@gmail.com

4thS. M. Jaybhaye
Assistant Professor in Information Technology
Sinhgad College Of Engineering
Pune, Maharashtra, India
smjaybhaye.scoe@sinhgad.edu