EMPIRICAL EVALUATION OF SHALLOW AND DEEP CLASSIFIERS FOR RUMOUR DETECTION

Akshi Kumar[1*], Vaibhav Singh [1], Tuba Ali[1], Saurabh Pal[1], Jeevanjot Singh[1]

[1]Dept. of Computer Science & Engineering
Delhi Technological University
**akshikumar@dce.ac.in*

Abstract. Rumour Detection has attracted a lot of attention from the research communities ever since there has been a boom in social networking sites like Twitter, Facebook, etc for spreading information and news at a very fast pace. We aim to analyze the various machine learning algorithms as well as deep learning algorithms on PHEME dataset which is a benchmark dataset and then propose an avant-garde method for detecting rumours based on deep learning architecture. In our work a detailed comparison of shallow classifiers as well as deep classifiers on the basis of performance parameters (accuracy, precision, recall) is done which provides a deeper insight into the field of rumour detection. Same goes for the comparison between the deep learning techniques. On application of machine learning algorithms on PHEME dataset, maximum accuracy of 78.54% is achieved using conventional statistical TF-IDF weighing. However, on application of deep learning algorithms using word embeddings, results improve drastically, with Bidirectional LSTM yielding the highest accuracy of 91.1 %.

Keywords: ***machine learning; deep learning; rumour detection; PHEME.***

1. Introduction

The unprecedented growth of social media [1, 2] and its outreach has led to a surge in unverified news/information which is often named as “rumour” [3]. Rumours affect our daily lives as well as businesses across the globe. In 2018, in India a spate of mob killings was sparked by a rumour about child kidnappings spread on social media platform Whatsapp[4]. In this day and age, when a powerful tool such as the Internet is easily accessible to nearly every person, it is the need of the hour to quickly detect rumour, trace its source and compute its veracity in order to provide non-falsifiable news to the general public. Rumours may be spread due to any individual or organization’s collective agenda. They can be used to insinuate election results, proselytize a biased opinion or may engender disaster to financial markets.

In the wake of such exploitation, there is scope for research to accurately and quickly detect rumours. The controversial information having unverified veracity status posted on social network sites like Twitter is identified and analysed by the rumor detection system in time-sequential format to warn users that the unverified information posted may turn out to be untrue[5]. In the research literature, rumour has been defined in a variety of ways. For example, rumour can be defined as “unverified and instrumentally relevant information statements in circulation” [6]. Oxford Dictionary defines rumour as “a currently circulating story or report of uncertain or doubtful truth”. The Merriam Webster dictionary defines rumour as “a statement or current report without known authority for its truth”. We stick to the definition of rumour as “a piece of circulating information whose veracity status is yet to be verified at the time of posting”. A rumour is nothing but a slice of information which may be represented in a textual or graphical way whose veracity has not yet been verified therefore there is uncertainty on its truth value while it is in circulation in public domain [4, 7].

Twitter is chosen for our research work, since it is one of the most popular and eminent social media platforms from which a relevant dataset can be generated [8]. The benefit of choosing twitter dataset is that, messages are only limited to 280 characters, and largely involve text as opposed to other platforms like Facebook or Whatsapp, where images, graphical content, videos are shared as frequently as text. The authenticity of users is established by having an account verified by twitter. Such accounts are genuine and this provides a means to distinguish between fake accounts and statutory accounts. Even the users belong to different strata, for example, we have Regular users, Public figures, and company representatives who can simply like, comment or repost the tweet [9]. Users “tweet” their thoughts and ideas for many reasons. Some people tweet to state their opinions, to express their emotions over a subject, some do it for self-attention and some to promote their content or business. For a normal user, it becomes increasingly difficult to distinguish between true facts and unverified information. This makes Twitter a very conducive environment for spreading and cascading of rumours. Detection of rumours in the preliminary phase of rumour propagation is extremely crucial so that the destruction and havoc caused by rumours are minimized. Rumours can cause an unimaginable damage in various sectors of life, for example-

- **Opinions and Sentiments** [10-14] - religious views, ideological feelings can be easily distorted via any unverified piece of information over social media platforms. This can cause riots, distrust and suspicion within public in general
- **Business and Finance-** rumours can be extremely destructive for financial and economical areas of any country. A similar example is stated in the introduction. They can hamper the economic growth, can lead to significant spurt in inflation which coupled with other factors can lead to recession and job loss.

There has been a heavy dependency on journalists and manual annotators, who detect and classify information as rumour or non-rumour. The truthfulness of any source of information is often established on online websites where people come together and use their resources to do the needful. This suffers from a very serious drawback that many of the news reports might be accidentally excluded and there is a long delay in detection of rumours. This indicates that a real-time or online detection of rumours is a very difficult task. Hence, it is required to devise methods that help to automatically detect rumors on social media sites. Shallow classifiers employ statistical machine learning models like naive bayes, SVM, decision tree, random forest etc which perform satisfactorily on the numeric features, but fail to understand the the context of the text. Shallow classifiers suffer from this disadvantage that they can work only on numeric features. A proper scheme is henceforth required to extract or convert text into some numeric encoding. Further a proper assessment of the language is also needed since the text is composed in a particular language where words come in a finite and definite sequence. A random encoding will not give desired results. This is where deep learning techniques come into picture. A lot of advancements have been made in converting a sequenced text, in our case which is a collection of sentences building a tweet. Deep learning architectures [15] such as recurrent neural networks (RNN) have helped a lot in this direction. Natural language constructs such as TF-IDF [16] and bag of words have greatly boosted the performance. Further adding to the performance are LSTM, GRU, and BILSTM. A proper arrangement of textual data containing sequence of tweet has been processed in forward and backward mode simultaneously in our proposed Bidirectional LSTM model and thereby taking into consideration the significance of global context to facilitate twitter post in a timely fashion. Likewise, the deep classifiers when used with advanced techniques of word embeddings can more efficiently and effectively apprehend the contextual semantics of the information and make the rumor detection algorithm perform better.

To facilitate future research enthusiasts, we have provided an empirical study of shallow classifiers which encompasses the standard machine learning algorithms along with the deep learning techniques on the benchmark dataset provided by PHEME [17]. Further we describe in detail a new model to detect rumour based on bidirectional LSTM which falls under the category of deep learning. Coupled with deep learning techniques, we have employed word embeddings instead of TF-IDF method of converting textual features into mathematical representations. This work will help readers in giving a quick glance at the existing work in the vast field of rumour detection involving both shallow as well as in deep learning techniques as well as, the readers will be exposed with a new deep learning architecture to solve this ongoing problem.

The paper is divided into separate sections as follows: in section 2, the summary of the literature survey on rumour detection is explained. In section 3, we throw light on the working and description of the system architecture and the experimental design as to how data is prepared for the various classifiers, and how we compare various techniques. The novel approach of bidirectional LSTM is also presented in this section. In section 4, we present the results of model and validate the proposed method. In section 5, we conclude the paper by presenting the future scope with the present limitations.

2. Literature Survey

Zhifan, et al. [18] divided rumours into 3 broad categories, depending upon features collected from Twitter dataset. They ascribed to content, Twitter, and network. Similarly, Vahed, et al. [19] segregated the features accordingly and redeemed those microblogs which are rumours or supporting rumours. Both use shallow classifier techniques which form the basic machine learning algorithms. A slight variation in procedure is observed in Aker et al. [20], where author's bearing(stance) based on tweet is also taken into consideration in determining the position of the tweet as being a rumour or not. The dataset utilized is Rumour-Eval. The most comprehensive and elaborate work in the field of rumour detection is probably done by Zubaiga et al. [21], in which they provided a complete survey on rumour detection and resolution. Rumours have been classified as long standing and newly emerging rumours, depending on the extensiveness of the of transmission of the rumour. They have voluminously described the theoretical work for research on building rumour classification architecture, the datasets available, and future scope. They have also worked on a linear sequential technique (Conditional Random Field) to distinguish a post as rumour or non rumour [21]. They have utilised a combination of 'content-based' and 'social' features, which outperforms all the other classifiers in terms of precision and F1-score. Naïve Bayes was the best in terms of recall.

Castillo et al. [22] analyzed the credibility of news information which propagates through Twitter, but still relying on statistical machine learning algorithms (J-48). Apart from statistical classifiers, many people worked on deep learning techniques for rumour detection. Jing Ma et al. [23] in their work presented an RNN-based rumor detection model on Sina Weibo and twitter dataset. They utilized continuous variable length time series, and presented multilayer RNNs with hidden units for classification. They achieved 88.1% accuracy on twitter dataset and 91% on Sina weibo dataset with GRU technique and Another improvement in RNN architecture is done by Li et al. [24] by combining deep bidirectional GRU(gated recurrent unit) with normal RNN. Input sequence is considered in both the directions, first in forward and then in backwards along the timeline simultaneously, to achieve better results. They achieved 88.9% with 3 layered GRU. Another research done by Zhang et al. [25] is an addition to previous RNN models, there is an additional layer of autoencoder which also improves results in comparison to statistical machine learning algorithms. Furthermore, Zhang et al. [26] had also contributed by providing a simple RNN architecture for unsupervised learning based on users behavior on Sina-weibo dataset with an accuracy of 92.49% on Sina weibo dataset.

3. System Architecture

The typical rumour resolution task flow starts by first identifying whether a given piece or a given set of information is a rumour or not. This phase is termed as rumour detection. At this point the input is basically a raw chunk of data which is circulating in public domain and such a topic might or might not be a rumour. If the news event/report turns out to be true news, then it's automatically discarded since its veracity or truth value is known to us. But on the other hand if the classification of the information turns out to be a rumour, then this data is fed to the next phase which is called as rumour tracking. This part of the whole architecture will collect and cluster out relevant topics, discussions, and posts related to the identified rumour. Fundamentally a rich vocabulary or in other terms, a wide array of corpus is built around the particular rumour. This huge bunch of information is passed to the next phase which is called as rumour stance identification or rumours polarity identification. After collecting all the relevant discussions, the sentiments, and polarities for each of the input, are determined. This objective step segregates the input as question oriented, a positive orientation, a negative orientation or a neutral one, depending upon the subject polarity and various natural language structures. Lastly, we have the veracity determination, which will finally predict that whether the given rumour at first place was true or false.

Our research aims at the first component of this elaborate architecture, where we determine whether a given chunk of information is a rumour or non rumour. Formally this can be defined as a binary classification task in the language of supervised learning. The following fig.1 depicts the system architecture.

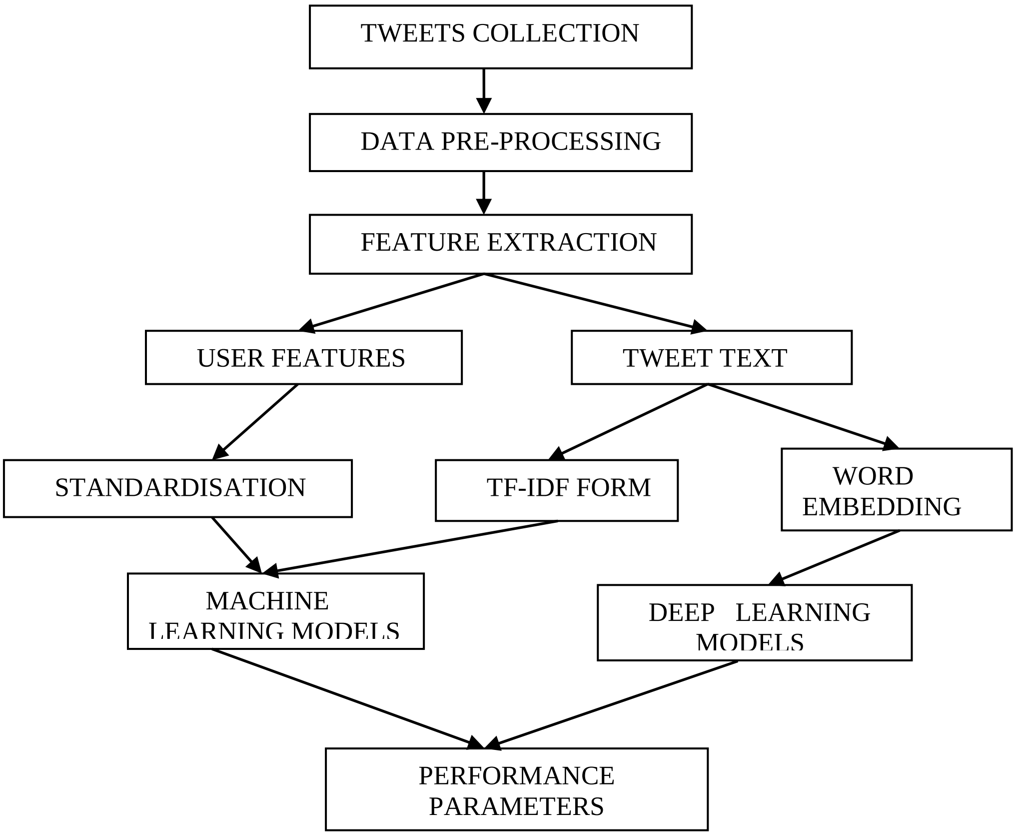

Fig.1. System Architecture

The following sub-sections discuss the details of the system architecture

3.1. Dataset

The initial step in the workflow is to retrieve the desired data from Twitter. For this purpose we have used PHEME dataset [17]. This dataset is openly available for enthusiastic researchers across the globe. This dataset is manually annotated and contains set of finite tweets about five news topics. Such a dataset can be used for research purposes for experiments on rumour detection. The data set was developed by the University of Warwick in conjunction with Swissinfo, part of the Swiss Broadcasting Company [27]. Journalists, working in Swissinfo collaborated with researchers from Warwick and developed this dataset. The dataset contains 5 events each of which contains a finite number of tweets, which are either rumours or non-rumours. The tweets are described by a set of features or attributes, for example Table 1 shows the set of features that describe the source tweet. Each of these features is inscribed in a nested JSON format.

Table 1. Summary of PHEME dataset.

News topic	Rumours	Non-rumours	Total
Charlie Hebdo	458	1621	2079
Ferguson	284	859	1143
Germanwings Crash	238	231	469
Ottawa Shooting	470	420	890
Sydney Seige	522	699	1221

3.2. Data Pre-processing

The next step is of data pre-processing [8]. This is done by making the dataset free of any noise, missing values, and inconsistencies. We remove hashtags (#), special symbols with no relevant meanings or context. Stop words and emoticons are also removed from the text. Tokenization and stemming are performed. After preprocessing we get a total of 5802 tweets, which are categorized into 5 different topics as shown. Each news topic contains a fixed percentage of rumour and non-rumours also shown in Table 1. Each of the tweet is characterised by a variety of features such as text, number of status, number of friends, screen name, tweet count, etc. some of which have been shown below in Table 2. The dataset is converted to CSV (Comma separated values) files for training and testing since python is used for implementation of shallow and deep learning models. We first get rid of unwanted features like country_code, utc_offset etc.

- We then perform a two step cleaning process, in the first step we clean the user features and in the second step we clean the textual features. The user features such as number of status, friends count, share count, etc are cleaned by replacing NULL values with 0 and normalizing them between 0 and 1 so as to reduce the effect of outliers.
- The textual features namely tweet is cleaned by removing emojis, and hashtags and any other special character such as space, @, !. word tokenization is performed followed by removal of delimiters followed by stopping and finally removing those words which are not present in English dictionary.

Table 2. Features and their description

Features	Description

Text	Denotes the content of the tweet
Followers count	Indicates the numeric quantity of number of followers of a user
Friends count	Indicates the number of friends of the user
Created at	Denotes the time of creation of a tweet
Name	Indicates the name of the user
Verified	Indicates whether the user is verified or not
Status _count	Denotes the number of the status of the user

3.3. Feature Extraction

The numerical features after cleaning and normalization are used as such but the tweet features are extracted by first forming a relevant textual representation. In our work we have used two such representations: TF-IDF for shallow classifiers and word embeddings for deep classifiers.

● **TF-IDF** (Term Frequency-Inverse Document Frequency)- It is often used in text mining and information retrieval, this is a further extension of Bag-Of-Words technique. This method helps in calculating the relative significance of a word to the respective document. It is a statistical measure which increases proportionally to the number of times a word appears in the document and is balanced or neutralized by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. Mathematically its defined as

$W_{i,j} = tf_{i,j} * \log(N/df_i)$
where $tf_{i,j}$= number of occurrences of i in j
df_i =number of documents containing i
N=total number of documents

● **Word Embeddings**- To understand the dense representation of words and their meanings, we have used word embeddings in our paper. Word Embeddings generally tries to map a word using a dictionary to a vector. Word embeddings can be learned from textual data. It is same as applying a Neural Network on textual data present in the dataset. Each word in the dataset is converted into a unique integer value. The technique that we have used in our word embeddings is skip gram technique. We have trained our embeddings instead of using pre trained embeddings like GloVe or Word2vec.

3.4. Shallow Classifiers

Shallow classifiers can generate good generalized predictive model with only a few layers of composition. This work focuses on a comparative analysis of six well-known shallow machine learning algorithms: decision tree, SVM, KNN, Random forest, SGD and Naive Bayes. The classifiers are briefly described below

● **Decision Tree (DT) -** It is a categorical classification of dataset which takes the continuous and discrete variables as input and thereby producing the output based on categories specified. Since, we are working on the labeled dataset and hence it is a good choice to run our data on it and analyzing the results.

- **Support Vector Machine (SVM)-** It is a supervised learning model (also called discriminative classifier) defined by separating hyper-planes. That is, given labeled dataset, it outputs an optimal hyper-plane which categorizes the new data to be analyzed.
- **K nearest neighbors (KNN)-** KNN is one of the most basic yet essential classification algorithms in Machine Learning. prior data (also called training data), is classified according to the distance parameter from its neighbors and hence divided the tweet as rumour or non-rumour depending upon the textual information and other features.
- **Random Forest (RF)-** Random forest algorithm is a supervised classification algorithm. This algorithm creates a forest with a number of decision trees and thus optimizing the results of decision tree.
- **Stochastic Gradient Descent (SGD)-** SGDClassifier is a simple yet efficient approach to discriminative learning of linear classifiers (SVM, logistic regression, a.o.). This estimator implements has been applied widely in large-scale and sparse machine learning problems often countered in text classification and natural language processing. Therefore, we have taken SGDClassifier into consideration to analyze textual information.
- **Naive Bayes (NB)-**This classification algorithm is used for predictive modeling. Bayes Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge and analyses that how the presence of a particular feature in our dataset is related to the presence of any other feature. Since, rumor posts heavily depend on the prior data available about that post and also the characteristics related to that post.

3.5. Deep Classifiers

- **Multi layer perceptron -** They form the simplest of the deep learning techniques. The structure is extremely simple and basic in nature. Primarily there is an input layer and an output layer which performs the decision or classification depending upon the weights given to each neuron, which takes the input. Each layer is represented as simple neurons. In our work we have used hidden layers, which take inputs from previous layers, and pass the output to next layer. Since a single layer would amount to a shallow classifier, hidden layers are added to make it deep. At each phase the activation function decides the output to the next layer.
- **Recurrent neural network (RNN) -** They are a special kind of neural network which are also the part of 'feed forward class of neural networks'. They are powerful and robust owing to their internal memory which facilitates in remembering things about input they receive, thus being precise in predicting what's coming next. Hence, it was easy for us to consider the time sequence dataset. They are used especially in variable-length sequential information because they can form a much deeper understanding of a sequence and its context, compared to other neural network architectures.

- **Gated Recurrent Unit (GRU)-** Introduced by Cho, et al.[28] in 2014, GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results. The architecture can be decomposed into 2 main gates, a) update gate, b)reset gate.
- **Long Short Term Memory (LSTM):** The structure of an LSTM has the only distinction that, the repeating module does more operations enabling it to remember long-term dependencies. The architecture can be decomposed into 3 main gates, a) update gate, b)output gate, c)forget gate. Since these gates are conjoined they can acquire information over a considerable period of time. This avoids the problem of vanishing gradient. The biggest disadvantage with the usage of regular RNN is that they only utilize the backward context of the input or feed sequence and completely ignore the future context or context at timestamp ahead.
- **Bidirectional LSTM (BILSTM)-** Bidirectional LSTM resolve this shortcoming by handling the data in two directions, first in a regular left to right or feed forward direction and then in a reverse direction that is from right to left. These two directions from the hidden layers but they behave as a separate unit. These independent units are combined and fed to the output layer. The Bidirectional LSTM introduced by Hochreiter & Schmidhuber called Long Short-Term Memory [29, 30] is applied on the dataset which we have used in our model to detect rumours. It outperforms feed-forward networks as shown in the section VI, because in sequential data where context becomes important, the latter models do not take this into consideration. In case of rumour detection, given a tweet when looking at the words comprising the tweet, the sequence model tries to derive relations with the other words of the same sentence. It takes into account both the forward and backward direction of the sentences or in another word, past event and future events both are taken into account which in generality gives a global context and helps in finding the real meaning of the tweet. This drastically improves the performance of the model.

We have applied 4 deep learning algorithms on the mentioned dataset. We have taken 2 layers of Bidirectional LSTM. For optimization on our dataset, Adam Optimizer has been used on the model. Mathematical ReLU function [31] has been used as activation function. We have trained the model till a minima is achieved for the loss function. Dropout of 0.2 is used to avoid over-fitting. The following table 3 depicts the hyper-parameters set for the Bidirectional LSTM model.

Table 3. Parameter values for our Bidirectional LSTM model

Hyper-parameters	**Value**
epochs	35
Activation function for input and hidden layers	ReLU
Activation function for output layer	Softmax

Loss function	categorical__crossentropy
Dropout value	0.2
Optimizer function	Adam Optimizer(learning rate=0.001)

4. Results and Discussion

The results discussion is divided into two parts: for shallow classifiers and deep classifiers. For the shallow classifiers, the results are discussed using accuracy, precision and recall as performance parameters whereas for deep classifiers accuracy is used as the performance parameter. The following sub-sections presents the results.

4.1. Shallow classifier performance

The results of applying the 6 machine learning algorithms on the five news topics are tabulated as decimal fractions, presented in the following table 4. On application of these machine learning algorithms, we observe that random forest presents the best results with an accuracy of 77.4% in ferguson To avoid overfitting, 80% of the dataset is used for training and the rest 20% is used for testing [32]. Best results are obtained using such a ratio of training and testing data. Particularly results are shown for ferguson news topic and parameters used to evaluate different classifiers are accuracy, precision, and recall. Fig.3 depicts the accuracy of shallow classifiers graphically.

Table 4. Performance results of Shallow classifiers

Techniques	accuracy	precision	recall
KNN	71.23	54.56	53.42
RF	78.54	66.61	58.18
SGDC	77.61	38.93	50.68
SVM	76.37	42.65	53.22
NB	77.12	39.78	49.32
DT	74.59	55.24	58.43

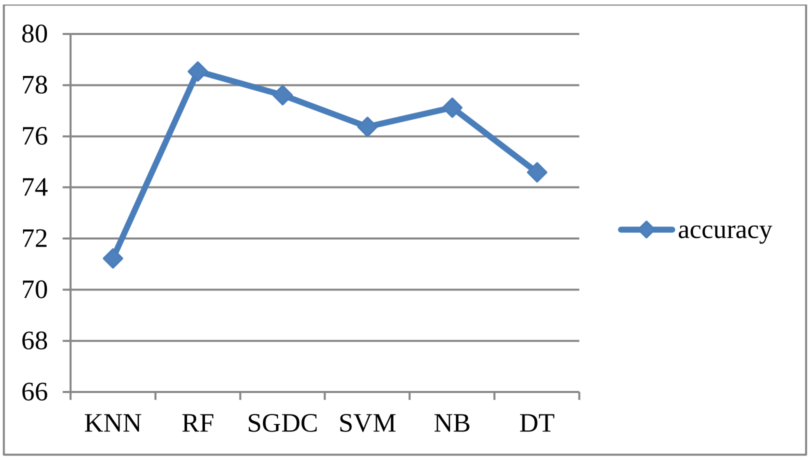

Fig.2. Accuracy of shallow classifiers

5.2 Deep Classifier performance

Fig. 4 shows the comparative analysis of various deep learning techniques which have been used in previous works and our proposed model. The proposed BILSTM model outperforms all with an accuracy of 91.1%. For reference and comparison we have compared a 3 dense layered perceptron, simple RNN, single layer LSTM, GRU and lastly our novel approach of Bidirectional LSTM. Compared to other architectures considered so far, Bidirectional LSTM is the most powerful with an accuracy of 91.1% when only 2 layers are included 85.6% by simple RNN followed by a 2.8 percent improvement via double layer LSTM architecture. The problem of over-fitting on the less available dataset caused the accuracy of BILSTM to decrease drastically when the number of hidden layers is increased.

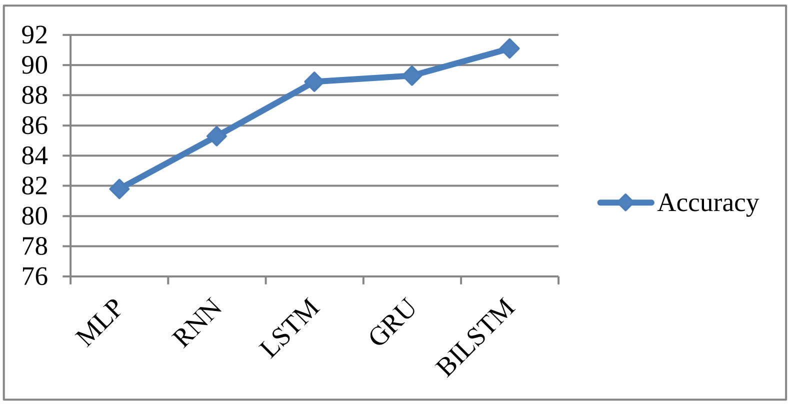

Fig.3. Accuracy of deep classifiers

The following fig.4 depicts the comparison of shallow and deep learning model accuracies.

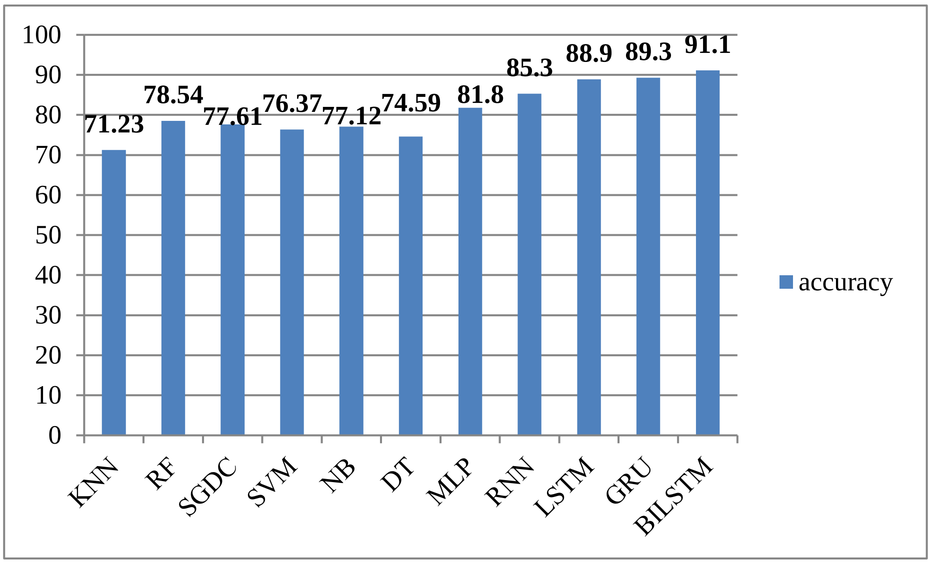

Fig.4. Performance comparison of shallow and deep classifiers

5. Conclusion

This paper empirically contrasted the six supervised learning algorithms, namely, Naive Bayes, Support Vector Machine, Decision Tree, K- nearest neighbor, Stochastic Gradient Descent, and Random Forest on PHEME dataset with 5809 tweets related to five events and the results were evaluated for the classifier performance, based on precision, recall and accuracy. The best accuracy and precision is achieved using Random Forest, followed by Support Vector Machine (SVM) followed by Naïve Bayesian, k-nearest neighbor, neural network and SGD classifier which demonstrated the lowest accuracy and precision. We would like to mention that in shallow classification holds the state-of-the art method which is conditional random fields and utilized various user and content features and have worked upon time sequence by incorporating hidden markov model on which conditional random fields are based.

Further in this paper, we introduced a novel approach to handle the problem of rumor detection based on deep learning architecture, that is, Bidirectional LSTM. A lot of experiments had been done in deciding the number of hidden layers. On the basis of many trials, it can be concluded that a double layer Bidirectional LSTM gives the best results. The work of this paper considered the text features of tweet and employed them in deep learning architecture, by forming word embeddings which gave better results than TF-IDF model for vocabulary formation. Since the data was limited to 5802 tweets, the results cannot be generalized to real-time rumour detection. Future work will include studies to fuse deep hidden features of text with other statistical features. Also, most of the rumour detection research has been done on textual data, whereas in real-time, websites like Facebook, Twitter, Whatsapp generate multimedia and graphical data, which can be studied for further research on rumour detection.

References

[1] A Kumar, R Khorwal, S Chaudhary, 2016, A survey on sentiment analysis using swarm intelligence, Indian J Sci Technol 9 (39), 1-7

[2] MPS Bhatia, A Kumar, 2010, Paradigm shifts: from pre-web information systems to recent web-based contextual information retrieval, Webology 7 (1), 1

[3] A. Kumar, S. R. Sangwan, 2018, Rumour Detection using Machine Learning Techniques on Social Media, International Conference on Innovative Computing and Communication, Lecture Notes in Networks and Systems, Springer

[4] A. Kumar, S.R Sangwan, & A. Nayyar, 2019, Rumour veracity detection on twitter using particle swarm optimized shallow classifiers, Multimed Tools Appl (2019). https://doi.org/10.1007/s11042-019-7398-6\

[5] Z. Zhao, P. Resnick, and Q. Mei. 2015. Enquiring minds: Early detection of rumours in social media from enquiry posts. In Proceedings of the 24th International Conference on World Wide Web. 1395–1405.

[6] N. DiFonzo and P. Bordia. 2007. Rumour, gossip and urban legends. Diogenes 54, 1 (2007), 19–35.

[7] A. Zubiaga, G. W. S. Hoi, M. Liakata, R. Procter,. "Detection and resolution of rumours in social media: A survey." arXiv preprint arXiv:1704.00656(2017).

[8] A Kumar, A Jaiswal, 2017, Empirical study of twitter and tumblr for sentiment analysis using soft computing techniques, Proceedings of the world congress on engineering and computer science 1, 1-5

[9] B. Liu, Sentiment Analysis Mining Opinions, Sentiments, and Emotions. Chicago, Cambridge University Press, 2015.

[10] Kumar, A., Dogra, P., Dabas, V., 2015, Emotion analysis of Twitter using opinion mining International Conference on Contemporary Computing (IC3),IEEE, 285-290.

[11] Kumar A., Jaiswal A. (2019). Systematic literature review of sentiment analysis on Twitter using soft computing techniques. Concurrency ComputatPractExper. https://doi.org/10.1002/cpe.5107

[12] Kumar A, Garg G. (2019). Systematic literature review on context-based sentiment analysis in social multimedia. Multimed Tools Appl. https://doi.org/10.1007/s11042-019-7346-5 (In-press)

[13] Kumar A, Sebastian TM. (2012a). Sentiment Analysis: A perspective on its Past, Present and Future. International Journal of Intelligent Systems and Applications. 4(10):1-14.

[14] Kumar A, Sebastian TM. (2012b).Sentiment analysis on Twitter. IJCSI International Journal of Computer Science Issues. 9(3):372-378.

[15] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar and M. Abdel-Basset, "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network," in *IEEE Access*, vol. 7, pp. 23319-23328, 2019. doi: 10.1109/ACCESS.2019.2899260

[16] Bhatia MPS, Kumar, A. (2008). A Primer on The Web Information Retrieval Paradigm. Journal of Theoretical & Applied Information Technology. 1;4(7).

[17] Zubiaga, Arkaitz; Wong Sak Hoi, Geraldine; Liakata, Maria; Procter, Rob (2016): PHEME dataset of rumours and non-rumours. figshare. Dataset.

[18] Yang, Zhifan, et al. "Emerging rumour identification for social media with hot topic detection." Web Information System and Application Conference (WISA), 2015 12th. IEEE, 2015.

[19] Qazvinian, Vahed, et al. "Rumour has it: Identifying misinformation in microblogs." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[20] Aker, Ahmet, L. Derczynski, and K. Bontcheva. "Simple Open Stance Classification for Rumour Analysis."arXiv preprint arXiv:1708.05286 (2017).

[21] A. Zubiaga, G. W. S. Hoi, M. Liakata, R. Procter, "Exploiting context for rumour detection in social media." International Conference on Social Informatics. Springer, Cham, 2017.

[22] C. Castillo, M. Mendoza, B. Poblete, "Information credibility on Twitter," Proc of the 20th International Conference on World Wide Web. New York: ACM Press, 2011, pp. 675-684.

[23] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. Wong, M. Cha. "Detecting Rumors from Microblogs with Recurrent Neural Networks." In IJCAI-16, pages 3818-3824 , 2016

[24] L. Li, G. Cai, N. Chen: A Rumor Events Detection Method Based on Deep Bidirectional GRU Neural Network" in 2018 3rd IEEE International Conference on Image, Vision and Computing

[25] Y. Zhang, W. Chen, C. Kiat Yeo, C. T. Lau, B.S. Lee: "Detecting Rumors on Online Social Networks Using Multi-layer Autoencoder" in 2017 IEEE Technology & Engineering Management Conference (TEMSCON

[26] Y. Zhang, W. Chen, C. K. Yeo, C. T. Lau, B. S. Lee: " Unsupervised rumor detection based on users' behaviors using neural networks" in 2017 Elsevier B.V.

[27] A. Zubiaga, G. W. S. Hoi, M. Liakata, R. Procter, and P. Tolmie, "Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads," PLoS ONE, pp. 1–33, 2015. [Online]. Available: http://arxiv.org/abs/1511.07487

[28] K. Cho et al, "Learning Phrase Representations using RNN Encoder–Decoder". Available: https://arxiv.org/pdf/1406.1078v3.pdf

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997

[30] A. Graves, N. Jaitly, A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM", Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU), pp. 273-278, Dec. 2013.

[31] Y. Li, Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation", *Proc. Adv. Neural Inf. Process. Syst.*, pp. 597-607, 2017.

[32] A. Kumar, TM Sebastian, 2012, Machine learning assisted sentiment analysis, Proceedings of International Conference on Computer Science & Engineering (ICCSE'2012), 123-130.