Automatic Twitter Rumour Detection using Machine Learning

Devarsh Patel
Dept of Information Technology
MPSTME, NMIMS University
Mumbai, India
devarshpatel2605@gmail.com

Nicole D'Souza
Dept of Information Technology
MPSTME, NMIMS University
Mumbai, India
nicolemichelledsouza@gmail.com

Riddhi Gawande
Dept of Information Technology
MPSTME, NMIMS University
Mumbai, India
riddhi.v.gawande@gmail.com

***Abstract*—Information generation and its dissemination increases day by day on a very large scale as the count of users increase on social media. These platforms are a stage for the people to exchange their ideas and opinions. Social media microblogging platform (ex. Twitter) is the go-to place in case of discussion about any important event. Information spreads at a lightning pace on twitter. This leads to rapid spread of false information i.e. rumours which can cause a feeling of unrest among the people. Hence, it is crucial to analyze and verify the degree of truthfulness of such content. The automatic detection of rumours in its initial stages is a challenge because of the complexity of the text. In this paper, we have implemented and compared different existing machine learning algorithms on the PHEME dataset to identify and detect the rumours. The performance of the models has been analyzed.**

Keywords— Rumour Detection, Machine Learning (ML) , Twitter, Social Media, PHEME.

I. INTRODUCTION

Social Media platforms like Instagram and Twitter have become a stage for people to be vocal about their opinions and ideas. These platforms not only provide a stage for people to be vocal but also help influence and have an impact on the perspectives of other people. Especially a platform like Twitter which is a powerful and famous microblogging platform which caters to a large audience. Microblogs limit the content size and is useful to spread news and notifications quickly to a large group of people. This generates large amount of data every second which makes it increasingly necessary to monitor the content on social media [1]. Microblogging leads to the spread of false information and news which severely affects the people's perception, behaviour, government actions and the state of a situation. This quick spread of rumours questions the accuracy and credibility of any information on the social media sites [2]. It sometimes also leads to ignorance of honest news. In a situation of emergency, rumours lead to additional chaos and causes unrest among the people, thus making the matters worse. There have been reported incidents of spread of wrong information at large which is a threat to the people as well as the government. It causes financial and sentimental damage. Hence, it becomes essentially important to separate the true news from rumours. Detection of rumour has gained increasing importance in recent times because of the amount of fake information that is being spread on a daily basis. It has become imperative to identify the rumours or unverified information at the initial stage and stop its dissemination by taking counter actions like either correcting the source or reporting the source to the admin, etc. This would help avoid unnecessary controversies and chaos. It is difficult to detect a new rumour at an early stage because of lack of training to the model about the topic.

In this paper we study the results of several machine learning models on PHEME dataset. In Section II, we will discuss about the related works. Section III discusses the methodology that has been used and the results obtained. In Section IV, we cover the conclusion, as well as the future scope of the project.

II. RELATED WORKS

In [1], 10 algorithms have been implemented on 2 different datasets of COVID-19 for unigram and bigram data. They did a comparative study of the various classifiers. In the Covid-19 dataset, the MLP algorithm gave the best f1-score for the unigram model, whereas the SVM algorithm showed the highest f1-score for the bigram model. For the Arabic dataset, the performance of KNN is superior to that of any other classifier that uses unigram or bigram models. Authors in [2] first rank different types of features based on their category into either User based (a description of the qualities of the person who wrote the tweet, supported by the argument that individuals who spread rumours have comparable features) or Content based (converting the information included in the tweet into a form that can explain its text). They then implement Random Forests, Naive Bayes, and Support Vector Machines with a combination of different characteristics. They achieved 78% accuracy for an event.

Authors of [5] incorporate the content and user-based functionality of the extracted tweets with TF-IDF. They use 13 features over 7 machine learning algorithms and they also implement CNN algorithms. CNN and machine leaning models (Logistic Regression and Random Forest classifier) have similar F1 score but the recall of CNN is better than the two Machine leaning algorithms. The F1 score before applying the CNN model was ranging from 0.46 to 0.76, after CNN it came to an average of 0.77. The best-case scenario came from a combination of logistic regression and CNN. [6] makes a claim that information propagated i.e. tweet-based features are much more important than user-based research. They confirm this by achieving an increased accuracy of 87.9% after removing the attributes that contribute the least for classification. Their J48 decision tree classifier gives the highest accuracy, and they propose an algorithm to find the original source of the tweet.

Authors in [7] provide an answer to the issue of lack of performance of the machine learning models against unseen rumours. They analyse the association between 13 different machine learning models, characteristics, and the rumours that have been discussed. They then suggest an ensemble or a set of

978-1-6654-9291-1/22/$31.00 ©2022 IEEE

models in order to get the best possible outcomes. When it came to identifying unknown rumours, an ensemble of machine learning models consisting of Random Forest, XGBoost, and Multilayer perceptron achieved an F1 score of 0.79, which was higher than any single machine learning model. They came to the conclusion that using a collection of various models to identify hidden rumours was more effective than using any of the other tactics. In addition, they recommend avoiding the usage of DT, LDA, QDA, and MLP in favour of concentrating on user-based characteristics. And suggests that one should steer clear of NB, LDA, and SVM when considering characteristics that are dependent on propagation. For the purpose of modelling credibility at the tweet level, Nguyen et al. [11] developed an ensemble approach that is dependent on characteristics (namely, the authenticity of every and the entire event). They demonstrated that the suggested model achieves an accuracy of more than 80% in the first hour, which increases to greater than 90% as more time passes. However, they did not take into account how the performance varies when applied to a variety of rumour subjects and/or events.

On the PHEME dataset, the authors in [8] present a hybrid model that makes use of CNN and a filter-wrapping optimised Naive Bayes Classifier. The results demonstrate that the suggested CNN+IG-ACONB rumour classifier achieves a better F1 of 0.732. Their method simply makes use of text-based characteristics, but meta-features like retweet count and user-based variables may be learnt independently to construct a more robust model for debunking rumours. The enquiry-based rumour detection method was introduced by Zhao et al. [9], however their results stated the performance based only on precision, which only obtained 0.52 overall. In addition, the answer is dependent on their already being inquiry about the supposed incident i.e., rumours may not be detected if there are no sufficient queries about the rumour.

According to Aker et al. [10], writing is categorised according to the author's point of view (stance taken by the author which may be inferred by reading the text). They used the Pheme dataset in addition to the RumourEval dataset (generated from the Pheme dataset). Various classifiers such as Random Forest, KNN, etc were implemented in this process. The authors make use of a broad variety of characteristics, some of which are brown clustering, bag of words, and part of speech tagging. They also suggest a collection of additional problem-specific characteristics, which they refer to as the "AF-Feature Set," which enhances the performance of the system. One creates a cumulative vector consisting of all of the characteristics. On the RumourEval dataset, the Random Forest classifier achieves the highest level of performance, outperforming Turing with an accuracy rate of 79%. On the Pheme dataset, the greatest performance came from the J48 Decision Tree. In addition, the elimination of the 'AF-Feature Set' results in a reduction of roughly 2.5% in the accuracy.

Giasemidis et al. [12] performed a research to detect rumours using a variety of ML algorithms. They did this by extracting 87 features from a dataset that contained 72 rumours that were spread on Twitter and classifying those features into one of three categories: message-based, user-based, or network-based. Decision Tree gave the best result with a F1-score of 0.97.

III. METHODOLOGY

Figure 1 shows the methodology that has been followed while implementing this research. Firstly, PHEME data set was taken for research purpose. This data was then cleaned and pre-processed for further analysis. Then the machine learning algorithms are implemented and evaluated.

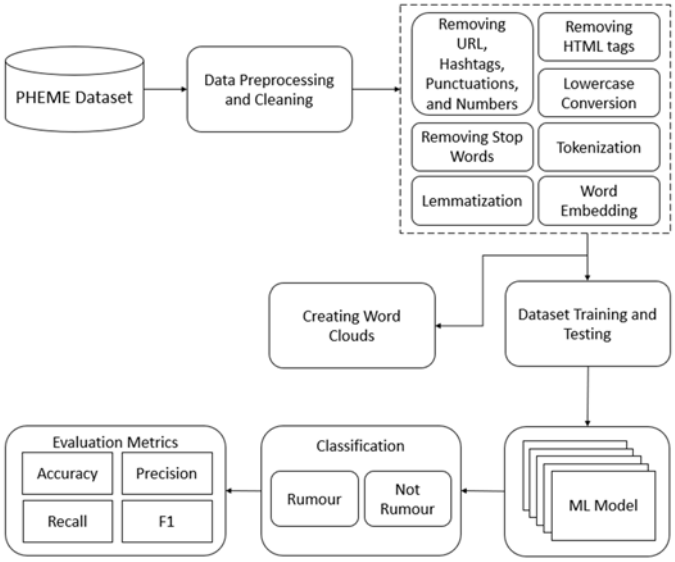

Fig. 1. Methodology

A. Dataset

PHEME dataset was used in order to train and test the models. It is a labelled dataset which is available to the public. It is a set of tweets of 5 different global events with the label as rumour or non-rumour and it also contains the source tweet. 1500 rumours and non-rumours each were taken from all the 5 events making a combined dataset of 15,000.

B. Data Preprocessing

The processes of data pre-processing and data cleaning are very necessary for the development of machine learning models. Text pre-processing is an important part of natural language processing which helps in the detection of rumours. It helps transform the text that is provided into a format that can be readily comprehended by the algorithms. Data Cleaning steps included:

1) **Removing URL, Punctuation, Hashtags and Numbers:** They do not play a factor in deriving wehther a given sentence is rumour or not. Hence, it is essential to get rid of numbers, commas, fullstops etc since they do not affect the result.

2) **Removing HTML tags:** The dataset was in JSON format so the extracted tweets contained HTML tags which had to be removed.

Data pre-processing involves the following:

1) **Lowercase conversion:** It is more efficient to use a single representation for words that have the same meaning and spelling as opposed to having several words for each situation. For instance, Violent and violent mean the same. Therefore, converting all uppercase letters to lowercase letters is necessary in order to make the word vector proportionate.
2) **Removing stop word:** NLTK stop-words are very commonly found in most rumours so we decided to make use of a count and tf-idf vectorizer to count the number of occurrences of a word and evaluate whether it is a stop word or not.
3) **Lemmatization:** This was done over stemming because however done, lemma i.e. the root word always belongs to an actual word which is not the case in stemming which usually just crops the end of the word i.e. suffixes to obtain the stem. For instance, stemming the word 'Caring' would return 'Car'. However, lemmatizing the word 'Caring' would return 'Care'. We used spacy lemmatizer, which is faster than its counterparts NLTK and TextBlob.
4) **Tokenization:** It refers to the process of separating a sequence of textual input into individual tokens that have meaning. For this model we have used tweet-tokenizer which was designed especially for tweets which makes them better equipped tokenizing words.
5) **Word Embedding:** This technique represents the word mathematically We used TF-IDF for word vectorization, it helps in determining the mathematical significance of words by using statistical measures.

C. Machine Learning Models

The PHEME dataset was split into 70% for training and 30% for testing. We implemented 12 machine learning models, as stated below in Table. 1. On further analyzing Table 1, it is observed that SVC and LSTM performed the best with accuracies, 90% and 89% respectively. The also had very close F1-scores. Support Vector Classifier is a supervised machine learning method that is often employed for classification tasks. The SVC algorithm works by mapping data points to a high-dimensional space and then separates the data into two according to the the most optimal hyperplane for dividing the data. Multinomial Naive Bayes is learning technique in Natural Language Processing (NLP) based on Bayes theorem. Using the Bayes theorem, the program forecasts the tag of a text, like an email or a news article. It analyses the chance of each tag being associated with the supplied sample and returns the tag that is associated with the highest probability.

Table. 1. Results (NR: Not Rumour, R: Rumour)

Sr No	**Algorithm**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			NR	**R**	**NR**	**R**	**NR**	**R**
1	SVC	90%	0.90	0.89	0.89	0.90	0.90	0.90
2	Random Forest	86%	0.85	0.87	0.87	0.85	0.86	0.86
3	Decision Tree	83%	0.86	0.81	0.79	0.87	0.82	0.84
4	Ada Boost	83%	0.82	0.85	0.85	0.81	0.83	0.83
5	Multinomial Naive Bayes	88%	0.88	0.89	0.89	0.88	0.89	0.88
6	Passive Aggressive	86%	0.85	0.86	0.86	0.85	0.86	0.86
7	KNN	68%	0.73	0.64	0.55	0.80	0.63	0.71
8	LSTM	89%	0.88	0.91	0.92	0.87	0.89	0.89
9	MLP	85%	0.86	0.84	0.84	0.86	0.85	0.85
10	Logistic Regression	88%	0.88	0.89	0.89	0.87	0.88	0.88
11	Gradient Boosting	71%	0.65	0.87	0.93	0.50	0.76	0.63
12	XGB Boost	69%	0.64	0.83	0.90	0.49	0.75	0.62

KNN and XGB Boost performed poorly with accuracies of 68% and 69% respectively. XGB Boost has the least F1-Score of 0.69. The remaining models had accuracy between the range of 71-88%. K-Nearest Neighbor is an algorithm for Supervised Learning. The K-NN algorithm assumes that there is a similarity between the new case or data and the previous cases, and it places the new case in the most pertinent category. K-NN can be utilized for both Regression and Classification but is primarily utilized for Classification. XGBoost is a strategy for machine learning that makes use of a gradient boosting framework and is centered on the concept of decision trees. Artificial neural networks tend to do better at making predictions with unstructured data than all other algorithms and frameworks.

D. Word Cloud

Figure 2 and 3 are word clouds indicating the most commonly used words in tweets that are labelled as rumours and not-rumours.

Fig. 2. Rumour Word Cloud

Fig. 3. Not-Rumour Word Cloud

IV. Conclusion and Future Work

The models in this paper outperform most papers mentioned in the literature survey. This could be attributed to the dataset being balanced i.e. each instance has equal number of rumour and non-rumour rows.

These days, it's impossible to imagine life without social media; today's youth rely on it not only for keeping in touch with friends but also for learning about and discussing a wide range of topics, which in turn shapes how they see the world. Although the propagation of misinformation is not a recent phenomenon, its rate of dissemination has recently increased. This study aims to provide methods for identifying and countering rumours, which are often spread deliberately to sway public opinion or mislead readers. For the purpose of this study we focused more on the content of the tweets rather than the users sending it, utilizing 3000 rows from each event in the PHEME dataset giving us a total of 15000 rows. We trained our model on 12 different algorithms and obtained the highest accuracy of 90% with the Support Vector Classifier.
However there is a lot that can be done in the future, including but not limited to

1) Traditionally, bilingual societies communicate in a mixture of English and their local language. Using natural language processing tools to manage code-mixed tweets can enhance a model's capacity to detect rumours.
2) A social network graph that traces the origin of the rumour can also be used to check the veracity of the tweet, and if multiple rumours can be linked back to the same account, the social media handle can temporarily suspend or banned.
3) Tweets may be classified as news, rumours, discussions, etc. This would help classify and distinguish tweets that may include rumours.
4) The majority of research focuses on identifying rumours from existing data, but very little research exists on identifying the legitimacy of tweets/comments streamed directly from social media platforms in real time.

Acknowledgment

We would like to acknowledge and express our sincere gratitude to our professor, Dr. Ashwini Rao (Assistant Professor - IT Department, MPSTME, NMIMS University) for her constant guidance and encouragement.

References

[1] N. Ashraf, H. Nayel and M. Taha, "A Comparative Study of Machine Learning Approaches for Rumors Detection in Covid-19 Tweets," 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 2022, pp. 384-387, doi: 10.1109/MIUCC55081.2022.9781707.

[2] A. Vijeev, A. Mahapatra, A. Shyamkrishna and S. Murthy, "A Hybrid Approach to Rumour Detection in Microblogging Platforms," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 337-342, doi: 10.1109/ICACCI.2018.8554371.

[3] M. Bharti and H. Jindal, "Automatic Rumour Detection Model on Social Media," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2020, pp. 367-371, doi: 10.1109/PDGC50313.2020.9315738.

[4] Sahana V P, A. R. Pias, R. Shastri and S. Mandloi, "Automatic detection of rumoured tweets and finding its origin," 2015 International Conference on Computing and Network Communications (CoCoNet), 2015, pp. 607-612, doi: 10.1109/CoCoNet.2015.7411251.

[5] Y. Kim, H. K. Kim, H. Kim and J. B. Hong, "Do Many Models Make Light Work? Evaluating Ensemble Solutions for Improved Rumor Detection," in IEEE Access, vol. 8, pp. 150709-150724, 2020, doi: 10.1109/ACCESS.2020.3016664.

[6] Kumar, A., Bhatia, M.P.S. & Sangwan, S.R. Rumour detection using deep learning and filter-wrapper feature selection in benchmark twitter dataset. *Multimed Tools Appl* (2021). https://doi.org/10.1007/s11042-021-11340-x

[7] Z. Zhao, P. Resnick and Q. Mei, "Enquiring Minds", *Proceedings of the 24th International Conference on World Wide Web*, 2015.

[8] A. Aker, L. Derczynski and K. Bontcheva, "Simple Open Stance Classification for Rumour Analysis," in *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*, 2017.

[9] T. Nguyen, C. Li and C. Niederée, "On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners", Lecture Notes in Computer Science, pp. 141-158, 2017.

[10] G. Giasemidis, C. Singleton, I. Agrafiotis, J. R. C. Nurse, A. Pilgrim, C. J. Willis and D. V. Greetham, "Determining the Veracity of Rumours on Twitter," in International Conference on Social Informatics, 2016.