CanarDeep: A Model for Rumour Detection in Benchmark PHEME Dataset

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

Submitted By:

AKSHAT SHRIVASTAVA

2K18/SWE/25

Under the supervision of

Dr. AKSHI KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

JUNE, 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

DECLARATION

I, Akshat Shrivastava, Roll No. 2K18/SWE/25 student of M.Tech (Software Engineering), hereby declare that the Project Dissertation titled "**CanarDeep: A Model for Rumour Detection in Benchmark PHEME Dataset**" which is submitted by me to the Department of Computer Science & Engineering , Delhi Technological University, Delhi in partial fulfillment for the requirement of the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi — Akshat Shrivastava

Date: 11-08-2020 — (2K18/SWE/25)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled *"***CanarDeep: A Model for Rumour Detection in Benchmark PHEME Dataset***"* which is submitted by Akshat Shrivastava, Roll No. 2K18/SWE/25, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment for the requirement of the award of degree of Master of Technology (Software Engineering) is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **(Dr. Akshi Kumar)**

Date: 11-08-2020 **SUPERVISOR**

Assistant Professor

Department of Computer Engineering

Delhi Technological University

ABSTRACT

A rumour is any statement that is not yet confirmed at the time of posting, irrespective of whether it's true or false. It is evident that rumours are an imperious threat to the credibility of the information providers. The sheer volume of information diffusion has led to an imperative need for questioning the tangibility of information. Unsubstantiated rumours on social media can cause significant damage by deceiving and misleading the society. It is essential to develop models that can detect rumours and curtail its cascading effect and virality. In this project, we proffer a CanarDeep model for rumour detection in the benchmark PHEME dataset. The proposed model is a hybrid deep neural model that combines the predictions of a hierarchical attention network (HAN) and a multi-layer perceptron (MLP) learned using context-based (text + meta-features) and user-based features respectively. A logical OR based decision-level late fusion strategy is used to dynamically combine the predictions of both the classifiers and output the final label as rumour or non-rumour. The results validate superior classification performance to the state-of-the-art. The model can facilitate timely intervention by buzzing an alarm to the moderators and further forming a cordon to inhibit the dissemination of spurious and junk content.

ACKNOWLEDGEMENT

I am most thankful to my family for constantly encouraging me and giving me unconditional support while pursuing this research.

I am extremely grateful to **Dr. Akshi Kumar** Asst. Professor, Department of Computer Science Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to her for the extensive support and encouragement she provided.

I also convey my heartfelt gratitude to all the research scholars of the Web Research Group at Delhi Technological University, for their valuable suggestions and helpful discussions throughout the course of this research work.

Akshat Shrivastava
(2K18/SWE/25)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

HAN	Hierarchical Attention Network
MLP	Multi-Layer Perceptron
POS	Part-of-speech
WHO	World Health Organization
UNICEF	United Nations Children's Fund
P	Precision
R	Recall
F1	F1 Score
GRU	Gated Recurrent Unit
ELMo	Embeddings from Language Models
CRF	Conditional Random Fields
GloVe	Global Vectors for Word Representations
LSTM	Long Short-Term Memory
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
AUC	Area Under the ROC Curve
ROC	Receiver Operating Characteristics

CHAPTER 1 INTRODUCTION

1.1. OVERVIEW

Social networking sites are platforms consisting of an abundance of information and news. With the meteoric advancements in social media, all sorts of information have become readily accessible for the public. The same piece of news or information may be reported by thousands or millions of people around the globe. There is a high variation in the information procured from these different sources. These variations lead us to believe that most of this information must have come from unverified sources. A statement is considered as a rumour if its current status is unverified. A surfeit of unverified information is publicized on social platforms daily. The sheer volume of information diffusion has led to an imperative need for questioning the tangibility of information. The ease in online account creation, posting accessibility, broad latitude and virality makes social media an ideal and seamless choice for perpetrators as they tend to hide behind fake or hacked profiles to spread gossip or misleading stories. The economics of social media too favors rumours, hate-speech, pseudo-news, alternative facts or fake news [1-3]. These rumours can leave a gargantuan impact on an individual or an organization as a whole. The ramifications of such a spread can lead to a societal epidemic.

Formally, a rumour is defined as “any piece of information put out in public without sufficient knowledge and/or evidence to support it thus putting a question on its authenticity” [4]. It spreads like wildfire and is believed overtly especially during a crisis. The wave of misinformation and rumour pertaining to the Covid-19 on social media and other digital platforms is a testimony to this rising infodemic. The petrifying part of Covid-19 related misinformation isn’t the folks who believe it, but the lack of virtuous information to counter it. Governments and many leading world health providers list out guidelines to evaluate information found on social media by advising use of trusted sources, such as official government or health care websites and their social media channels; (i) evaluating meta information available such as, inclusion of links and media, to assess reliability and (ii) searching other credible resources to see if they are sharing similar information.

To ensure information credibility, many social media platforms such as Facebook, Facebook-owned WhatsApp and Twitter have invested in strategies and tools dedicated to identifying rumours and improving online accountability. These follow obligatory

regulations or standard guidelines and rely on a combination of artificial intelligence, user reporting, and content moderators to implement rubrics for reliable and apposite content filtering. But the strategies and code of practices are opaque to the users whereas the moderators are overwhelmed by the sheer volume of content and the ordeal that comes from sifting through vexing posts. Moreover, the problem with rumours is that its virality is much faster than its debunking. Often, despite debunking a rumour, a re-posting of the same claim emerges. Thus, automated debunking of rumours and combating their viral spread is the need of the hour.

Formally, rumour detection is defined as "determining if a story or online post is a rumor or non-rumor (i.e. a real story, a news article)." A typical rumour analysis task consists of four components:

(1) *Rumour Detection:* where potential rumours are recognized.

(2) *Rumour Tracking:* monitors the tweet, filters and captures related posts.

(3) *Stance Classification:* determines the orientation of user's view as "in favour" or "against".

(4) *Veracity Classification:* knowledge is garnered based on the selection of significant features and subsequent classification is done to determine the actual truth value of the rumour.

In this project, we propose a model for the first component, i.e., the recognition of potential rumours. The remainder of this chapter sets out the research objectives and presents an outline of this thesis.

1.2. RESEARCH OBJECTIVES

The research proffers a novel hybrid model for rumour detection. The proposed *CanarDeep* model is a hybrid of Hierarchical Attention Network and Multi-layer perceptron that combines information from two sub-networks to detect & classify rumourous posts in real-time data. The model derives its nomenclature from the French word *"Canard"* which means 'unfounded, groundless or false report or story' and *deep* learning techniques applied in this model to detect rumours and combat its viral read. A primary approach to decipher the truth value of a post is to look for some user-based and text-based evidence. Some meta-features such as re-post count, and morpho-syntactic (exclamations) & typographic (capitalization, quotes) markers can also serve as non-trivial cues. Therefore, in the mix-fusion CanarDeep model, rumour detection is done by the individual classification model, namely HAN and MLP using context-based features

(textual + meta-features) and user-based features respectively. The context-based features are consolidated into a single feature vector using concatenation based early fusion. The HAN utilizes a bi-directional GRU with attention mechanisms at both word and sentence level. On the other hand, the MLP generates its output based on the user profile features using the back-propagation algorithm and adjusting the weights of the neurons accordingly. MLP is a simple yet highly sought-after model when it comes to binary classification using discrete numerical features [5]. To finally detect the post as rumour or non-rumour, a late fusion using a Boolean OR operation is done which combines and categorizes the output on the basis of two truth values. The performance of the *CanarDeep* model is validated on the PHEME benchmark dataset. The results are compared against state-of-the-art conditional random field classifier [6]. The main target is to achieve a high efficacy in rumour detection tasks with the help of our proffered model.

1.3. ORGANIZATION OF THESIS

The project report has been divided into five chapters. Each chapter deals with one component related to this thesis. Chapter 1 being introduction to this thesis, gives us the brief introduction about the project, thereafter chapter 2 tells about the literature survey which further includes related work section. Following up is chapter 3 which tells about the proposed work. Chapter 4 provides us with the experiments and results followed by final chapter, chapter 5, which is the conclusion of the thesis.

CHAPTER 2 LITERATURE SURVEY

The chapter explains various kinds of rumour and the work done so far in the field of automatic rumour detection.

2.1. TYPES OF RUMOUR

A rumour is any information put out in public without sufficient knowledge and/or evidence to support it. It is misleading, either intentionally or unintentionally. Rumours can be classified as true, false or unverified based on their veracity. The rumours belonging to the 'true' category are those which started off as unverified and later turned out to be true whereas those belonging to the 'unverified' category are the ones whose veracity remains unconfirmed. The rumours pertaining to the 'false' category are further classified into three types, namely, mis-information, dis-information and mal-information.

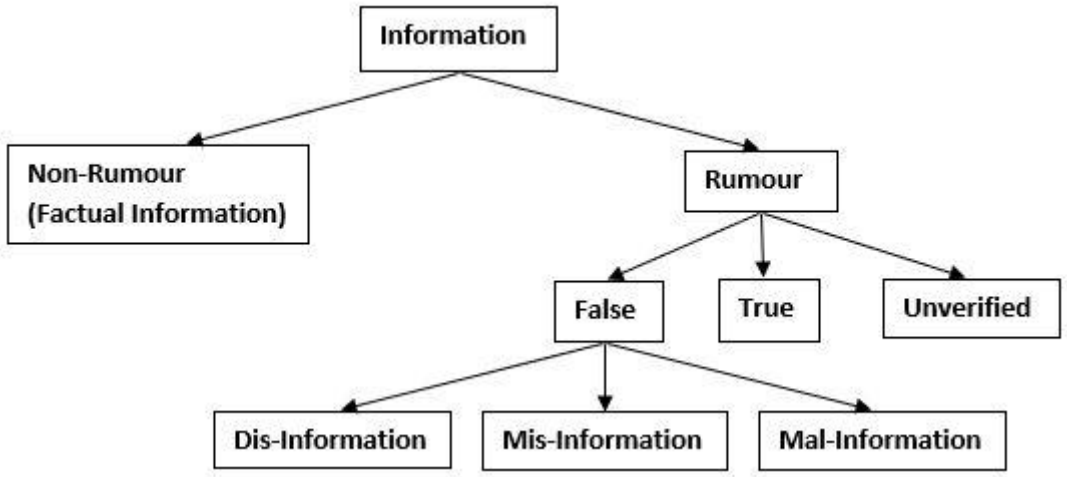


Fig. 2.1 Types of Rumour

The newfound social media landscape for communication, disseminating information and voicing opinions brings to us substantial risks of fabricated information. Much of the discourse on 'online information fabrication' conflates the above-mentioned three notions: misinformation, dis-information and mal-information.

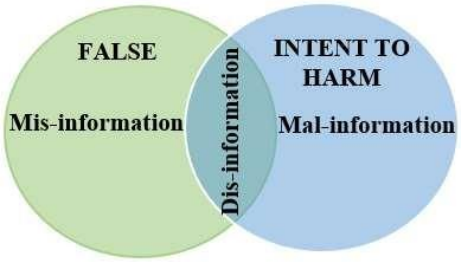


Fig. 2.2 The 'information disorders' in social media

These vary in accordance to the truth value of the content and the intent of information being created, produced or distributed (Fig. 2.2). That is, dis-information contains outright lies with no element of truth and is deliberately created to harm a person, social group, organization or country. Comparatively, in misinformation though the information is false, but it is not created with the intention of causing harm, rather it is an erroneous mistake. Mal-information is grounded on reality but either taken completely out of context or manipulated, with malicious intent to inflict harm on a person, organization or country. Undeniably, these 'information disorders' [7] that affect the social web have exposed us to the relentless virtual transgressions of lies, falsehoods and hate-crimes on the Web.

Table 2.1: Examples of rumours during some recent events.

Event	Rumour	Fact	Category
Coronavirus Pandemic (Covid-19)	In January 2020, UNICEF provided some information on the coronavirus which stated, "If the virus is exposed to a temperature of about 26 or 27-degree Celsius, it will be killed as it does not live in hot regions."	UNICEF did not make any such claims pertaining to the coronavirus. Also, WHO is responsible for issuing guidelines and advisories related to the coronavirus, not UNICEF.	Mis-Information
Death Reports of Kim Jong Un	In April 2020, the supreme leader of North Korea went missing and rumours related to his health were rife. There were reports suggesting he was "recovering from heart surgery at his compound in Wonsan, had contracted the coronavirus or was under quarantine despite reports in the North Korean media that there are no cases in the country." Some reports even said that he was dead.	According to Korean Central News Agency (KCNA), "Mr. Kim was accompanied by several senior North Korean officials for the ribbon cutting ceremony at the opening of a fertilizer factory on 2nd May, 2020." The people who were attending the event "burst into thunderous cheers of 'hurrah!' for the Supreme Leader who is commanding the all-people general march for accomplishing the great cause of prosperity", KCNA says.	Mal-Information

2.2. RELATED WORK

Social media grants the power to propagate information irrespective of whether the source is verified or not. This creates a risk of widespread rumours among the populace. Automatic rumour detection is essential, keeping in mind the volume and velocity of user-generated information on social media. This task becomes even more relevant with the exponential rise in the percentage of social media users.

Various studies in pertinent literature attempt to organize the wide variety of approaches to rumour detection. Zubiaga et al. [3] presented a comprehensive survey with a distinction between detection, tracking, stance classification and veracity classification for rumours. Kumar and Sangwan [4] applied different machine learning techniques for rumour detection. Cao et al. [8] explored approaches based on hand-crafted features, propagation, and deep learning.

Microblogging platforms such as Twitter and Sina Weibo have emerged as a focal point of rumour detection research. A range of machine learning techniques have been applied to Twitter- and Weibo-based datasets for rumour classification. Takahashi and Igata [9] applied filtering using keywords and retweet ratio to detect rumours in post-disaster tweets. Yang et al. [10] used client- and location-based features to train an SVM-classifier for detecting rumours on Sina Weibo. Liu et al. [11] proposed a real-time rumour detection system for Twitter based on the premise that rumours may result in conflicting reactions from the users, which allowed detection of instances even with less than five tweets. Liu and Xu [12] used user-specific features to develop an information-propagation model to distinguish rumours. Wang and Terano [13] proposed a graph-based pattern matching algorithm to detect rumours based on both structural and behavioral properties. Zhao et al. [14] proposed a clustering technique combined with "enquiry phrases" which decided the classification of the cluster as rumour or non-rumour.

Deep learning techniques have been applied to microblogging data to automatically learn representation of rumour data. Ma et al. [15] applied RNNs to learn hidden representations of contextual information of posts from Twitter and Weibo. Chen et al. [16] presented a deep attention model based on RNNs. Soft attention on recurrence allowed the model to selectively learn the temporal context of sequential posts. Jin et al. [17] proposed a multimodal approach by fusing features based on image, text and social

context. Image features were incorporated with a combination of textual and contextual features obtained using an LSTM and together passed to RNN with attention mechanism. Nguyen et al. [18] focused on early detection of rumours by using a hybrid model based on CNN and RNN. CNN was used to extract high-level representations of the rumour-related tweets and the RNN was used to process the time series obtained by CNN.

PHEME [19] is a benchmark dataset containing a collection of rumours and non-rumours posted on Twitter during breaking news, with annotations by expert journalists. Alkhodair et al. [20] focused on detecting breaking news rumours using a LSTM-RNN network to learn representations for rumours. Most closely related to our work is Zubiaga et al. [19], who used the PHEME dataset to train a sequential classifier- CRF, to use the context learnt during an event. To improve upon the performance of the automatic rumour classification task, this research puts forward a late fusion model. The proposed deep neural model uses HAN and MLP for learning two distinct input feature types and classifies the post into rumour or non- rumour category. The advantage of late fusion is that using multiple and diverse classifiers provides significantly more information to arrive at an accurate decision. It helps avoid the curse of dimensionality and synchronization between different features.

CHAPTER 3 PROPOSED METHOD

The proposed hybrid deep neural model is an amalgamation of two different deep learning algorithms, namely, HAN and MLP. The hybrid model is used for the binary classification of real-time posts as rumours and non-rumours. Typically, in machine learning, three types of data fusion strategies are used: feature-level (early), model-level (medial) and decision-level (late) fusion. Early fusion involves concatenation of features from different sources to obtain a single feature vector, which is more discriminative than any of the input feature vectors. The medial fusion involves concatenation of high- level feature representations from different inputs and the late fusion involves fusion of predictions from different classifiers. In our proposed model, we use early fusion to combine the textual features with the meta-features (collectively called as context-based features). Late fusion is applied to combine the decisions of multiple classifiers, namely HAN and MLP, trained using context-based and user-based features respectively, to produce a final common decision. There are various common and straightforward methods to accomplish decision level fusion, such as using logical operators, votes or weighted majority. In this work, to finally detect the post as a rumour, a Boolean decision system with an OR operation is used.

To learn the context-based features, the HAN classifier consists of an embedding layer, encoders and attention layers. ELMo 5.5B model [21] is used as the word vector learning technique to seed the classifier. ELMo employs a deep, bi-directional LSTM model to produce word representations. The bi-directional GRU with attention layer is firstly used at word-level and then repeated at the sentence-level. Therefore, the HAN architecture consists of five layers, namely, the embedding layer, word sequence encoder, word- level attention layer, sentence encoder and sentence-level attention layer and finally produces a document vector for a single input, which is passed through a sigmoid activation function to get the final output as either rumour (positive class) or non-rumour (negative class). The MLP classifier is used to learn the user-based features. It does so by using the back-propagation algorithm and adjusting the weights of the neurons after every backward pass, thus minimizing the error as much as possible and finally attaining convergence. The output layer of the MLP consists of a single neuron with a sigmoid activation function, which generates the final output as either rumour (positive class) or non-rumour (negative class). We fuse the outputs gathered from the two classifiers using

the Logical OR operation to generate the final output for a particular post. Fig. 3.1 represents the architecture of the proposed *CanarDeep* model.

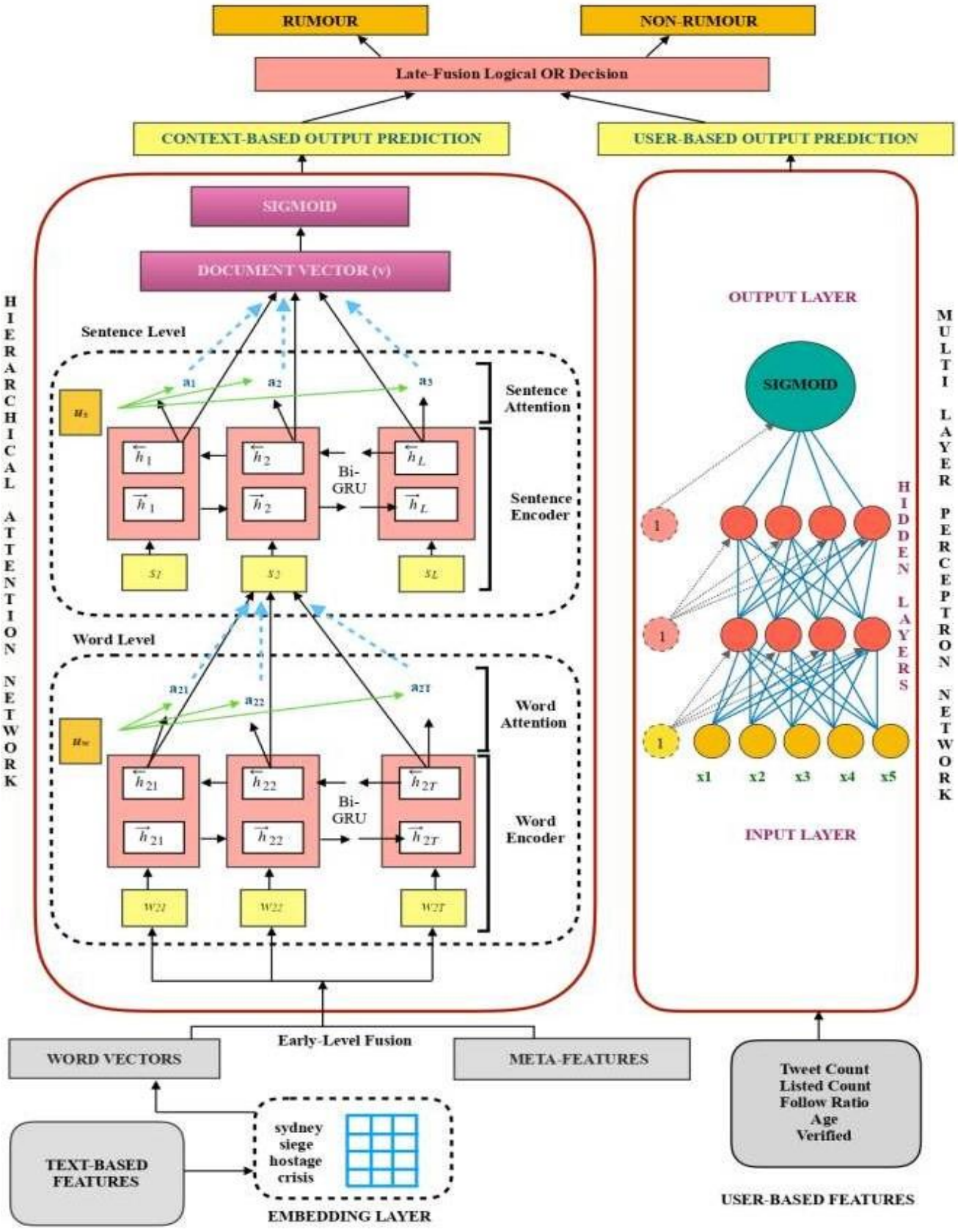


Fig. 3.1 The Proposed CanarDeep Model

3.1. PREPROCESSING

Preprocessing is the task of preparing the data in a manner, which is easier for the machine learning model to comprehend. The raw data is transfigured into clean data which is then used as input to the model. The following preprocessing of the features was done to make them suitable for the rumour detection task:

Data Cleaning: The data is cleaned by removing noise from it. Firstly, all the characters are converted to lowercase. Then, hyperlinks, unwanted characters, symbols and whitespace are removed from the text. Finally, stop words are removed from the text. Stop words are the most regularly occurring words in any language. These may be prepositions, conjunctions or interjections which are used very often in the text and do not really add to the meaning of the text.

Spell Check and Lemmatization: After the cleaning of data is done, spell check is done to correct any erroneous spellings. Then, lemmatization is performed on the text. Lemmatization converts each word to its base form by checking the lexicon, i.e., the root words obtained after lemmatization are morphologically correct. By using lemmatization, 'caring' gets converted to 'care' which would have been converted to 'car' using stemming.

Scaling: The user-based features undergo scaling. *StandardScaler* from the *sklearn* library for Python is used to scale all the user-based features. *StandardScaler* normalizes the features i.e. each column of the dataset, individually, so that each column/feature/variable will have mean = 0 and standard deviation = 1.

3.2. FEATURE EXTRACTION

Feature extraction is a vital task in supervised machine learning. In this work we have used context-based and user-based features to train the individual classifiers. Context-based features combine textual features and some content based meta-features such as word count, POS tags, capital ratio, etc. On the other hand, the user-based features comprise profile based discrete numeric features.

3.2.1. Context-Based Features: We have used a total of 11 context-based features that are input to the HAN classifier. These features consist of the textual content of the post as well as the supplementary details associated with the post. The description of these features is as follows:

- *Word Vectors*

  Word vectors are used to represent the relationship across words, sentences, and documents. They are simply the vectors containing numbers that show and map the meaning of the word for the model to understand. We built the word vector using the ELMo 5.5B model for word embeddings. We have trained five distinct ELMo 5.5B models, one model for each of the five runs, training with four events at a time, and using the remaining one for testing.

- *POS Tags*

  POS tags are used for grammatical tagging. They tag each word in the tweet with their respective grammar tags such as nouns, adverbs, adjectives, etc.

- *Capital Ratio*

  Most of the time, the word which has been spelled in capital letters tends to have more impact than the word written in lower case.

- *Word Count*

  Count of words in any particular tweet.

- *Use of Question Mark*

  Question marks sometimes represent the uncertainty of a saying, disrespect, impatience, or lack of tactfulness. Thus, it is necessary to have a binary feature that shows whether the tweet consists of question marks or not.

- *Use of Exclamation Mark*

  The exclamation marks in the tweets express surprise, astonishment, or any strong emotion resulting in additional emphasis. This binary feature represents the presence or absence of an exclamation mark in a tweet.

- *Use of Period*

  Punctuation might represent good writing and hence quality reporting. This binary feature represents the presence or absence of a period in a tweet.

- *Use of Colon*

  The use of a colon in tweets helps the user to add two independent clauses, thus allowing them to add two complete thoughts that stand alone as complete sentences. The presence of a colon may suggest careful reporting. This binary feature represents the presence or absence of a colon in a tweet.

- *Use of Comma*

  The use of a comma in a tweet suggests quality reporting. This binary feature represents the presence or absence of a comma in a tweet.

- *Favorite Count*

  This feature tells us how many people have marked a particular tweet as their favorite. The higher the count, the more are the chances of it not being a rumour because a higher favorite count shows that people believe in that tweet.

- *Retweet Count*

  This feature tells us how many people have retweeted a particular tweet. Retweeting is defined as the sharing of a tweet by a user so that the user's followers can also read the tweet. If the retweet count of a tweet is high, there is a good chance that the tweet is not a rumour because the users trusted it enough to share it.

3.2.2. User-Based Features: We have used 5 user -based features as input to the MLP classifier. These include:

- *Tweet Count*

  This feature depicts the count of tweets a user had posted on twitter.

- *Listed Count*

  This feature tells us the count of lists a user is a part of, i.e., the number of times they were added to a list by other users.

- *Follow Ratio*

  The reputation of a user is assessed on the basis of the count of their followers. But, sometimes, the count of followers does not reflect the true prominence of a user. For example, some users follow many others in order to be followed back. Keeping this scenario in mind, we take the follow ratio as a feature, which is the number of followers someone has divided by the number of people following them. It is basically the followers to following ratio.

- *Age*

  The age of a twitter user pertains to the years they have been using Twitter. It is the time from the setting up of the account to the time of the current tweet.

- *Verified*

This feature tells us if the user is verified by twitter or not. The verified users are less likely to spread rumours as compared to others.

3.3. HIERARCHICAL ATTENTION NETWORK

The first classifier that takes as input the context-based features is HAN. The HAN model was proposed by Yang et al. [5] for document classification. HAN is based on the fact that a document forms a hierarchical structure, i.e., words form various sentences, and then, those different sentences form the complete document. Firstly, the sentences are formed by representing the words in a vector form and then the document vector is constructed by using a vector form of those sentences. Thus, the HAN classifier tackles each post at word level and then again at sentence level, finally forming a document vector, leading to high efficiency in text classification tasks. To learn the context-based features, the HAN classifier consists of an embedding layer, encoders and attention layers. The encoders extract relevant context and the attention layers compute the degree of relevance of the sequence of tokens with respect to the document. The architecture consists of five layers, namely, the embedding layer, word sequence encoder, word-level attention layer, sentence encoder and sentence-level attention layer. The new age ELMo (Embeddings from Language Models) word embedding [21] is used as the word vector learning technique to seed the classifier for textual feature vector generation. The rest of the context-based features which are categorical in nature are concatenated to the word vector generated by the embedding layer as a feature-level fusion strategy. Then, the bi-directional GRU with attention layer is firstly used at word-level and repeated at the sentence-level. Finally, the HAN classifier forms a vector representation of the document and this document vector is then passed through a sigmoid activation function to generate the output as either rumour or non-rumour. The layers are described in detail next.

3.3.1 Embedding Layer: The embedding layer of a neural network converts an input from a sparse representation into a distributed or dense representation. Word Embedding facilitates natural language understanding by means of semantic parsing such that the meaning from text is extracted preserving the contextual similarity of words. In this research, we use the state-of-the-art pre-trained ELMo 5.5B word embeddings model [25] to generate the word vectors. We preferred ELMo over the conventional embedding models such as Word2Vec or GloVe, as ELMo offers contextualized word representations, which essentially means that the representation for each word depends on the entire context in which it is used. The same word can have two different vector

representations based on different contexts. ELMo creates vectors on-the-go by passing words through the deep learning model rather than having a dictionary of words and their corresponding vectors, as is the case with traditional word embedding models. Also, ELMo representations are purely character-based, which allows the network to form representations for words that are not seen in training. All this motivated us to use the ELMo 5.5B model for implementing the embedding layer. As mentioned before, the word vector generated using ELMo is concatenated with the meta-features comprehending a feature-level early fusion strategy.

3.3.2 Encoder: A GRU based sequence encoder is used in HAN. The GRU [22] uses a gating mechanism without employing any separate cells for memory to track the sequences of the state. It comprises of the following two gates: the update gate z_t and the reset gate r_t. The reset gate regulates the amount of contribution provided by the previous state to the current state. The update state z_t defines the extent of the past information to be added and also of the new information to be added to the current state. Both, the reset gate r_t and the update gate z_t jointly control the updation of information in the current state. At any time t, the new state h_t is calculated using (3.1):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}t. \quad (3.1)$$

where, h_{t-1} is the previous state and $\tilde{h}$ is the current state.

The update gate z_t is given as shown in (3.2):

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (3.2)$$

where, x_t is the sequence vector at time t.

The current state $\tilde{h}_t$ is computed as given in (3.3):

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \quad (3.3)$$

where, r_t is the reset gate.

The reset gate is computed as given in (3.4):

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (3.4)$$

3.3.2.1. Word Encoder: A mapping of discrete variables to a vector of continuous numbers is termed as an embedding matrix. Neural network embeddings prove to be utilitarian as they are able to lower the dimensionality of categorical variables and pertinently represent categories in the transformed space. Provided a sentence with words w_{it}, $t \in [0, T]$, we start by embedding the words to vectors with the help of an embedding

matrix W_e, $x_{it} = W_e w_{it}$. We summarize information from both the directions, namely, forward and backward using a bidirectional GRU in order to obtain annotations of the words. The forward GRU $\vec{f}$ reads the sentence s_i from w_{i1} to w_{iT} as given in (3.5).

$$\vec{h}_{it} = \overrightarrow{GRU}(x_{it}),\ t \in [1,T] \tag{3.5}$$

The backward GRU $\overleftarrow{f}$ reads the sentence s_i from w_{iT} to W_{i1} as given in (3.6).

$$\overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}),\ t \in [T,1] \tag{3.6}$$

The concatenation of the backward hidden state and the forward hidden state helps us in acquiring the annotation of the word w_{it}, i.e., $h_{it}=[\vec{h}_{it},\overleftarrow{h}_{it}]$.

3.3.2.2 Sentence Encoder:

Analogous to a word encoder, we utilize the sentence encoder to obtain the document vector from the given sentence vectors s_i, we employ a bidirectional GRU for encoding the sentence as given in (3.7) and (3.8):

$$\vec{h}_i = \overrightarrow{GRU}(s_i),\ i \in [1,L] \tag{3.7}$$

$$\overleftarrow{h}_i = \overleftarrow{GRU}(s_i),\ t \in [L,1] \tag{3.8}$$

To get the annotation of a sentence i we concatenated both, the backward hidden state h_j and the forward hidden state h_i, i.e., $h_i = [\vec{h}_i, \overleftarrow{h}_i]$.

3.3.3 Attention Layer: The hierarchical attention layers are used for the document level classification. Suppose the document L has sentences s_i and each sentence has T_i words. The notion of 'attention' is based on the fact the words in a sentence do not contribute equally to its meaning. Similarly, not all sentences contribute equally towards the overall meaning of the document. The word-level and sentence-level attention mechanism are described as:

3.3.3.1. Word Attention Layer: In trying to understand the meaning of the sentences, it becomes clear that all the words present in a sentence do not contribute equally to its meaning. Therefore, to find out those words which are key to the meaning of the sentences, we use word attention layer. The extracted words, then, form the sentence vector. To get the hidden representation of the h_{it}, we calculate u_{it}, using a one-layer MLP as given in (3.9).

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (3.9)$$

We, then, generate the normalized importance a_{it} through a sigmoid function to calculate the significance of the word as a similitude of u_{it} with a word level context vector u_w as given in (3.10).

$$a_{it} = \frac{\exp(u_{it}^T u_w)}{\Sigma_t \exp(u_{it}^T u_w)} \quad (3.10)$$

After that, we compute the weighted sum of the word annotations based on their weights which corresponds to the sentence vector s_i as given in (3.11).

$$s_i = \sum_t a_{it} h_{it}. \quad (3.11)$$

3.3.3.2. Sentence Attention Layer: Since, all the sentences in the document are not important to understand the meaning of the document, it is necessary to extract such sentences which are of more importance compared to the others. To get the hidden representation of h_i, we calculate u_i, using a one-layer MLP as given in (3.12).

$$u_i = \tanh(W_s h_i + b_s) \quad (3.12)$$

We, then, generate the normalized importance a_i through a sigmoid function to calculate the significance of the sentence as a similitude of u_i with a sentence level context vector u_s as given in (3.13).

$$a_i = \frac{\exp(u_i^T u_s)}{\Sigma_i \exp(u_i^T u_s)} \quad (3.13)$$

After that, we compute the document vector v_i which is the weighted sum of the sentence annotations depending on their weights. The summarization of all the information gathered from all the sentences of a document is present in the document vector v_i as given in (3.14).

$$v_i = \sum_i a_i h_i. \quad (3.14)$$

3.3.4 Document Classification: Finally, the classification of the document into one of the classes is done using the document vector v. The document vector v is made to pass through the last layer, i.e., the output layer, which uses a sigmoid function as we are dealing with a binary classification problem. This leads to the generation of the final output of the HAN sub-network.

$$p = \text{sigmoid}\ (W_c v + b_c) \quad (3.15)$$

The training loss is taken as the negative of the log-likelihood of the correct labels.

$$L = -\sum_d log\ pd_j \tag{3.16}$$

where, *j* stands for the label of document *d*.

3.4. MULTI-LAYER PERCEPTRON

The MLP forms the second classifier of our hybrid model, and it takes the user profile features as input. It can be thought of as a linear classifier, which means that it can segregate two different entities or classes from each other using a straight line. The input to a perceptron is usually a feature vector *x*, which is multiplied to a weight *w* and then finally added to a bias *b*.

$$y = w * x + b \tag{3.17}$$

A perceptron is a shallow neural network and thus incapable of solving classification problems in which the number of classes is more than two. It takes in a number of inputs and generates an output by forging a linear coalition by utilizing the weights of its inputs. It also, sometimes, passes the output through a non- linear activation function. This can be shown through an equation as follows:

$$y = \varphi(\sum_{i=1}^{n} wixi + b) \tag{3.18}$$

where, *w* stands for the weight vector, *x* stands for the input vector, *b* stands for the bias, and phi represents the non-linear activation function.

An MLP is made up of a number of perceptrons that are arranged in multiple layers (Fig. 3.2). It consists of an input layer, an output layer, and an arbitrary number of hidden layers.

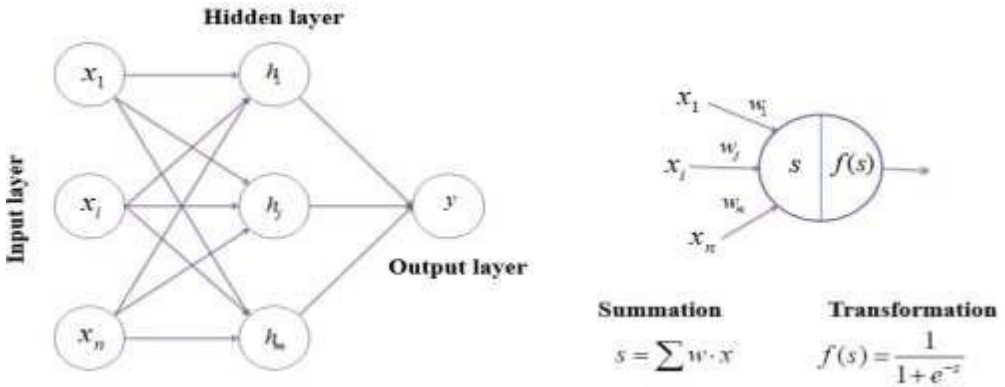


Fig. 3.2 MLP Architecture

In our MLP classifier, the input layer consists of five neurons, the two hidden layers consist of four neurons each, and the output layer has a single neuron with a sigmoid activation function. The user profile features are fed into the input layer of the MLP, and the final output is generated at the output layer as either rumour (positive class) or non-rumour (negative class).

3.5. DECISION LEVEL FUSION & FINAL CLASSIFICATION

The final output generation is done by employing an additional decision layer comprising the Logical OR operation. The fusion strategies of multiple input types can be categorized as model-free fusion and model- level fusion (medial). Model-free fusion can be further classified into early fusion (feature-level) and late fusion (decision-level). In early fusion, the different types of input features are firstly concatenated and then fed into a classifier, whereas in late fusion, the predictions of different classifiers trained for distinct input types are combined to provide us with the final output. Model-level fusion combines the advantages of both of these strategies by concatenating high-level feature representations from different classifiers. In the *CanarDeep* model, the output decisions from both the classifiers, i.e., HAN and MLP, are fused using the Logical OR operation. If the decision from either classifier is that the given input is a rumour, then the input is classified as a rumour. The given input is classified as a non-rumour if and only if both the classifiers decide that the input is a non-rumour. Table 3.1 illustrates the logical ORing.

Table 3.1: Decision Level Fusion using Logical OR

HAN	MLP	Decision
+ (Rumour)	+ (Rumour)	Rumour
- (Non- Rumour)	- (Non-Rumour)	Non-Rumour
+ (Rumour)	- (Non-Rumour)	Rumour
- (Non-Rumour)	+ (Rumour)	Rumour

The OR operation helps to debunk a rumour in with maximum possibility. If both the classifiers detect the post as rumour then it is irrefutably a rumour. Textual content and its meta-features provide valuable markers to indicate a rumour and therefore even if only the context-based classifier is indicative of rumour, the output is marked as rumour. This is because rumours are driven based on psychology and behavior of users which may alter with change in beliefs, confusion and anxiety or due to uncertainty. Hence, even a non-suspicious account can spread rumours. Likewise, if a user profile is identified

suspicious using the user-based classifier, it is also marked as rumour. The primary notion is that intelligent bots and masqueraded profiles tend to use professional services for believable content writing tactics which can often be missed by the context-based classifier. Thus, the ORing decides to discard a post only if both the classifiers classify it as a non-rumour.

CHAPTER 4 IMPLEMENTATION AND RESULTS

4.1. THE BENCHMARK PHEME DATASET

In this work, we have used a publicly available benchmark dataset for rumour detection tasks: PHEME. The PHEME dataset was introduced by Zubiaga et al. [19] in the year 2017. The dataset is annotated for two class labels, namely 'Rumour (0)' or 'Non-rumour (1)'. It is available in two versions: one with five events and the other with nine events. We have used the dataset with five events which consists of the event-based tweets related to Charlie Hebdo Shooting, Ferguson Unrest, Germanwings Crash, Ottawa Shooting, and Sydney Hostage Crisis, and annotated by expert journalists. The tweets are labeled for the support, certainty, and evidentiality of the rumour spread. The dataset has three levels of annotations: rumour stance classification, rumour veracity classification, and rumour detection. The event-wise labels within the dataset are given in Table 4.1.

Table 4.1: Labels for each event of the PHEME Dataset

Event	Rumour	Non-rumour	Total
Charlie Hebdo	458	1621	2079
Ferguson	284	859	1143
Germanwings Crash	238	231	469
Ottawa Shooting	470	420	890
Sydney Siege	522	699	1221
Total	**1972**	**3830**	**5802**

Two events have a class imbalance problem, where the count of non-rumours is considerably more than the count of rumours. To tackle this issue, we have evaluated the dataset in two different ways i.e., by evaluating individual events, and evaluating for the complete dataset. The strategies are discussed next:

- *Evaluating Individual Events*

Using the leave one event out approach, in which one event is used as a test set, while others are used for training. This is repeated 5 times so that each event is used as a test set once. This allowed us to mimic a real-time setting where if a totally new event comes up, the classifier is able to detect rumours from the knowledge gathered from events in the training set.

- *Evaluating the whole dataset*

By aggregating the output of all five runs as the micro-averaged evaluation across runs.

4.2. PERFORMANCE MEASURES

We have used the following evaluation metrics in our work:

Confusion Matrix:

It is a mapping of a relationship between what the model has predicted and what the actual result is supposed be as shown in Fig 4.1. If the predicted class is positive and actual class is positive as well, then we get the true positive section. If the predicted class is positive but the actual class is negative then we get the false positive section. Similarly, if the actual class is positive but the predicted class is negative then we get the false negative section and finally, if the actual class is negative and the predicted class is also negative, we get the true negative section.

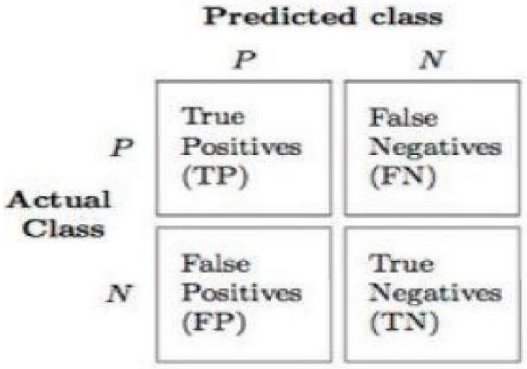


Fig. 4.1 Confusion Matrix

Precision:

It is the ratio of data elements that are correctly classified (for both the minority and majority class) to total number of classified instances.

$$P = TP / (TP + FP) \quad (4.1)$$

Recall:

The ratio of the minority class instances that are correctly classified to the total number of actual minority class instances.

$$R = TP / (TP + FN) \quad (4.2)$$

F-Measure:

Precision and Recall are used for performing the calculation of F- measure**.** It is calculated by taking the harmonic mean of Precision & Recall.

$$F\text{-}measure = 2 / (1/R + 1/P) \quad (4.3)$$

Parameter Values for HAN & MLP:

There are various parameters that have been used for both the sub-networks of our proposed model during the experiment. The values of those parameters for the HAN classifier can be seen in Table 4.2 below.

Table 4.2: Parameters used in HAN

HAN	
Parameter	**Value**
Embedding Dimension	300
Bi-LSTM Units	150
Hidden Units	300
Return Sequences	True
Trainable	True
Non-Linearity Function	ReLu
Loss Function	Binary Crossentropy
Optimizer	Adam
Dropout	0.5
Word Embedding	ELMo 5.5B
Batch Size	128
Epochs	7
Maximum Vocabulary Size	20000
Maximum Sentence Length	50
Maximum Sentence Number	5

The final parameter values for the MLP classifier can be seen in Table 4.3 below.

Table 4.3: Parameters used in MLP

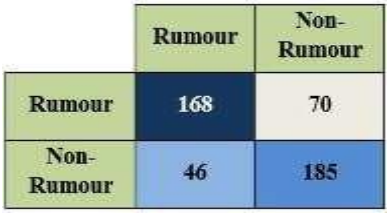
MLP	
Parameter	**Value**
Max Iterations	10
Solver	Adam
Learning Rate Initializer	0.01
Batch Size	200
No. of Hidden Layers	2
Units in each Hidden Layer	4
Tolerance	1e-4
Activation Function	ReLu

4.3. EXPERIMENTAL RESULTS

The performance of the *CanarDeep* model is examined with respect to each of the individual events to analyze how well the model performs across the dataset. A good understanding of how the proposed model performed on individual events can be obtained by taking a look at each event's confusion matrix. To compute the confusion matrices, we take a count of four values for each event:

- *True Positives* - number of rumours correctly identified
- *False Positives* - number of non-rumours that were incorrectly identified as rumours
- *False Negatives* - number of rumours that were incorrectly identified as non-rumours
- *True Negatives* - number of non-rumours correctly identified

The confusion matrices for each individual event are shown in Fig. 4.2 through 4.6, with actual class on the horizontal axis and predicted class on the vertical axis.

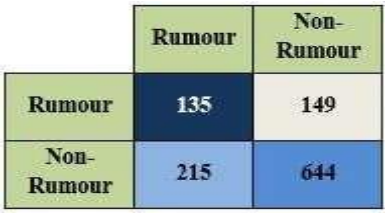


	Rumour	Non-Rumour
Rumour	168	70
Non-Rumour	46	185

Fig. 4.2 Germanwings

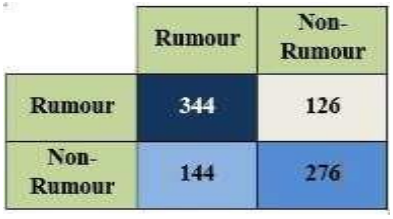
	Rumour	Non-Rumour
Rumour	410	112
Non-Rumour	240	459

Fig. 4.3 Sydney Siege



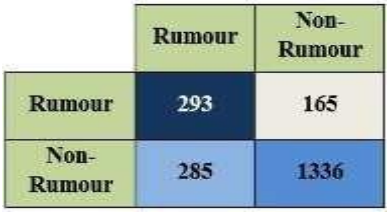
	Rumour	Non-Rumour
Rumour	135	149
Non-Rumour	215	644

Fig. 4.4 Ferguson



	Rumour	Non-Rumour
Rumour	344	126
Non-Rumour	144	276

Fig. 4.5 Ottawa Shooting



	Rumour	Non-Rumour
Rumour	293	165
Non-Rumour	285	1336

Fig. 4.6 Charlie Hebdo

Two out of five events in the PHEME dataset, namely, Charlie Hebdo and Ferguson, suffer from the class imbalance problem, while the other three i.e., Germanwings, Ottawa Shooting, and Sydney Siege do not. Data skewness or class imbalance proves to be a major hindrance in a classification task. In the case of class imbalance, the accuracy score is not used as an evaluation metric as it often leads to incorrect interpretation of performance. Thus, we have used F1 Score, Precision, and Recall as evaluation metrics to correctly represent performance on the PHEME dataset.

Other than precision, recall, and F1 score, we have also used AUC-ROC curves to judge

the performance of our model. AUC-ROC curve is an elementary metric for analyzing binary classification problems. It helps to understand the degree to which a learning model is able to distinguish between the given classes. The value of AUC ranges from 0.5 to 1. If the AUC is closer to 1, the model has a high potential for separating classes from one another. On the other hand, if the AUC is 0.5, the model has zero potential for segregating the given classes. The AUC scores of the five events present in the PHEME dataset range from 0.64 to 0.74. The ROC curve for each individual event is shown in Fig. 4.7 through 4.11.

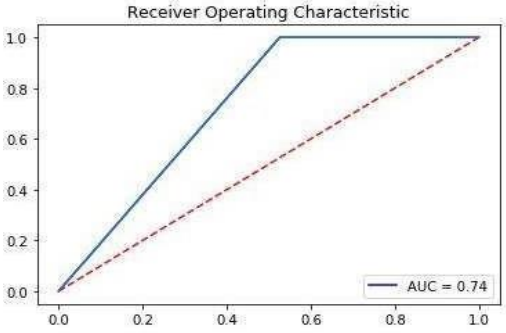


Fig. 4.7 ROC-Germanwings

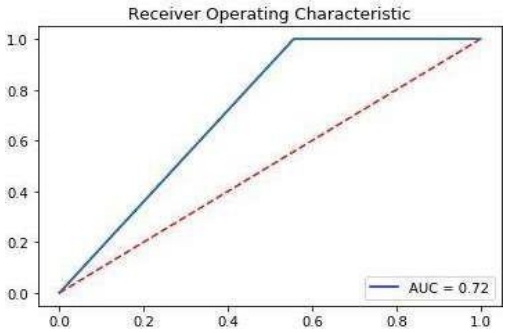


Fig. 4.8 ROC-Sydney Siege

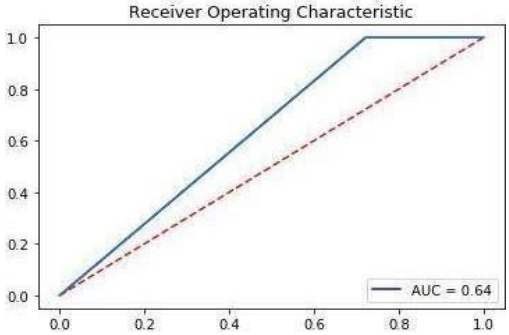


Fig. 4.9 ROC-Ferguson

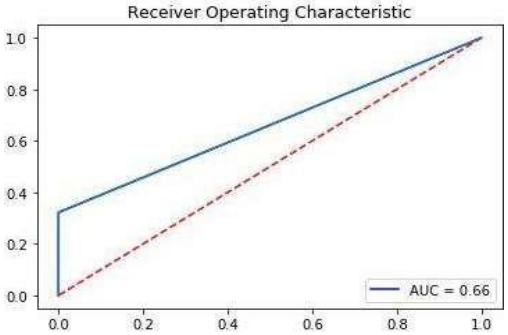


Fig 4.10 ROC-Ottawa Shooting

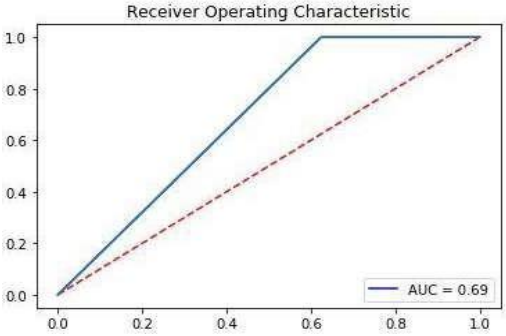


Fig. 4.11 ROC-Charlie Hebdo

From the AUC-ROC curves we see that the *CanarDeep* classifier did a marginally better job at separating rumours from non-rumours in Germanwings and Sydney Siege tweets

as compared to the other events.

We evaluate our model for individual events over five iterations with a leave-one-event-out approach. In each iteration, tweets related to one event are assigned to the testing set while those for the other four events are assigned to the training set. The experiments exhibit that our proposed model, which is a hybrid network, is able to handle multiple input types using the distinct classifiers to maximize the potential of each feature type. It achieves significant improvements over traditional methods of binary text classification. Our model performs fairly uniformly over the five individual events in the dataset, as shown in tables 4.4 and 4.5. The proposed *CanarDeep* model achieves higher precision compared to the state- of-the-art CRF classifier [6] across three out of the five datasets, the exceptions being Ottawa Shooting and Sydney Siege. The recall scores achieved by our model are also higher across four out of the five datasets, the exception being Charlie Hebdo. Our model also manages to strike an equilibrium between recall and precision, a qualitative improvement over the state-of-the-art. We also compare the results of the proposed model with an Attention-based Residual Network model used in [23].

Table 4.4: Classifier performance for Germanwings, Charlie Hebdo and Ottawa Shooting

	Germanwings			**Charlie Hebdo**			**Ottawa Shooting**		
Classifier	P	R	F1	P	R	F1	P	R	F1
CRF [6]	0.743	0.668	0.704	0.545	**0.762**	0.636	**0.841**	0.585	0.690
ARN [23]	0.704	0.702	0.703	0.677	0.711	0.693	0.680	0.678	0.679
CanarDeep [Proposed Model]	**0.753**	**0.755**	**0.754**	**0.732**	0.698	**0.715**	0.695	**0.696**	**0.695**

Table 4.5: Classifier performance for Sydney Siege and Ferguson

	Sydney Siege			**Ferguson**		
Classifier	P	R	F1	P	R	F1
CRF [6]	**0.764**	0.385	0.512	0.566	0.394	0.465
ARN [23]	0.719	**0.723**	**0.720**	0.574	0.583	0.578
CanarDeep [Proposed Model]	0.721	0.717	0.719	**0.613**	**0.599**	**0.606**

It is also worth mentioning that the proposed model does not let the datasets suffering from class imbalance hamper its performance, another qualitative improvement over the CRF classifier. CRF has an F1 score of 0.636 for Charlie Hebdo, whereas our model yields a score of 0.715 for the same. CRF did not provide satisfactory results for Ferguson

as well, with an F1 score of 0.465, which pales in comparison to *CanarDeep* model's score of 0.606 for the same. Our model followed a similar trend with datasets that are free from the class imbalance problem. The *CanarDeep* model performs better than CRF on all those three datasets, namely, Germanwings, Sydney Siege, and Ottawa Shooting, having F1 scores of 0.754, 0.719, and 0.695 respectively, as compared to those for CRF classifier being 0.704, 0.512 and 0.690.

Sydney Siege has the highest number of tweets among the events that do not have a class imbalance problem. The biggest improvement in the F1 score provided by our hybrid model is seen in Sydney Siege, with an improvement of 40.43%, as seen in table 4.5. It is worth noting that the *CanarDeep* model's overall superiority over the CRF classifier is because the former has performed exceptionally well in terms of recall score. Our model shows significant improvement in four out of the five events when it comes to recall scores.

The effectiveness of the *CanarDeep* classifier is assessed using the whole dataset, i.e., all the five events combined. The output of the five runs is aggregated by micro-averaging. We do this by appropriately combining the values of TP, FP, FN and TN for the individual events to calculate the value for precision, recall, and F1-score over the dataset as a whole.

Table 4.6: Classifier performance for the whole dataset

	Context-based features + User-Based features		
Classifier	**P**	**R**	**F1**
CRF [6]	0.667	0.556	0.607
ARN [23]	0.662	0.570	0.612
CanarDeep [Proposed Model]	**0.685**	**0.592**	**0.634**

We can see from the results in table 4.6 that with a mixture of user profile and context-based features used, and a mix of fusion strategies, *CanarDeep* outclasses CRF for all the three metrics. Our model gave a performance gain of 4.45% in terms of F1-score over the CRF classifier, solidifying our claim regarding our model's superiority.

CHAPTER 5 CONCLUSION AND FUTURE WORK

In this chapter, we firstly provide a brief conclusion of this work and then summarize the whole thesis. At last, we suggest possible future work in order to better tackle the problem at hand.

5.1 CONCLUSION

Rumours proliferate in times of crisis. The uncertainty and significance of the situation, combined with the lack of information fuels rumours in the virtual social world. It is thus imperative to question the tangibility of information. As a solution to debunk online rumours, this work proposed a novel *CanarDeep* model which combined information from two classifiers to detect & classify rumour in benchmark PHEME dataset. It took two inputs, namely context-based and user-based features, which were learned separately using the classifiers (HAN for context-based and MLP for user-based). The output predictions of these were then combined using a logical OR decision-level operation to categorize the post as rumour or non-rumour. The advantage of using early-level fusion to concatenate textual and meta-features as context-based features is that it does not isolate interactions between correlated features whereas the advantage of using decision-level fusion for final output is that the model need not synchronize between different types of features. The robustness of the technique is validated for both individual events and the whole dataset. The experimental evaluation reveals superior performance in comparison with the existing state-of-the art with a 4.45% gain in F1-score.

5.2 SUMMARIZATION

Our aim in this thesis is to detect whether a real-time post posted on social media is a rumour or not. The benchmark PHEME dataset is used for carrying out this study.

In chapter 2, we review the non-technical and technical studies dedicated to rumour detection. Spread of rumours is a societal epidemic phenomenon and is generating severe harm to people and organizations. The chapter deals with the kinds of rumours, the work done in this field and also the background studies that are important for performing the analysis. In this thesis, our target media object consists textual data along with some meta data about the texts as well as the users.

Chapter 3 illustrates the methodology proposed by us. We have utilized the

benchmark PHEME dataset and perform preprocessing and feature extraction on the data. The proposed CanarDeep model consists of two separate deep learning models, namely, HAN and MLP. HAN is used for text-based features whereas MLP is used for user-based features. The outputs from both the classifiers are combined together using a Logical OR operation to generate the final output. We further explained each of these classifiers in detail. The chapter also introduces all the features that are used as input for the CanarDeep Model.

Chapter 4 is where we show the implementation details, experimental setup and classification results. Here, we have explained the setting of various parameters that has been used for performing the experiments. We have defined the proper distribution of the data in the PHEME dataset. Furthermore, we have analyzed our model individually for each of the five events in the PHEME dataset as well as for all the five events combined. The results are compared with the existing state-of-the-art CRF classifier and we observed that our proposed CanarDeep model outperformed the existing state-of-the-art with a 4.45% gain in F1-Score.

5.3 FUTURE SCOPE

Although our approach encompasses a vast number of rumours and non-rumours, our experiments conducted on the PHEME dataset limit us to tweets that have been retweeted for a minimum of 100 times. A classifier that is evolved enough to identify tweets with the potential to be most retweeted, at an early stage, would allow the detection of rumours on time. Likewise, experimenting with a twitter dataset wherein the retweet count does not act as a roadblock in the detection of rumours would prove helpful in taking our work further ahead. Also, as more recently, the country-specific content written in native language is also compounding the linguistic challenges in rumour detection. The future work in rumour detection warrants a new line of inquiry to address these challenges.

APPENDICES

APPENDIX 1: LIST OF PUBLICATIONS (COMMUNICATED)

CanarDeep: Rumour Detection in Benchmark Dataset using Hybrid Deep Neural Model

Akshi Kumar[1*], Akshat Shrivastava[2]
[1,2]Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
**akshikumar@dce.ac.in*

Abstract. Unsubstantiated rumours on social media can cause significant damage by deceiving and misleading the society. It is essential to develop models that can detect rumours and curtail its cascading effect and virality. In this paper, we proffer a *CanarDeep* model for rumour detection in benchmark PHEME dataset. The proposed model is a hybrid deep neural model that combines the predictions of a hierarchical attention network (HAN) and a multi-layer perceptron (MLP) learned using context-based (text + meta-features) and user-based features respectively. A logical OR based decision-level late fusion strategy is used to dynamically combine the predictions of both the classifiers and output the final label as rumour or non-rumour. The results validate superior classification performance to the state- of-the-art. The model can facilitate timely intervention by buzzing an alarm to the moderators and further forming a cordon to inhibit the dissemination of spurious and junk content.

Keywords: Rumour; Deep learning; HAN; MLP; Early fusion; Late fusion

1. Introduction

The newfound social media landscape for communication, disseminating information and voicing opinions brings to us substantial risks of fabricated information. Much of the discourse on 'online information fabrication' conflates three notions: misinformation, disinformation and mal-information. These vary in accordance to the truth value of the content and the intent of information being created, produced or distributed (Fig.1). That is, dis-information contains outright lies with no element of truth and is deliberately created to harm a person, social group, organization or country. Comparatively, in misinformation though the information is false, but it is not created with the intention of causing harm, rather it is an erroneous mistake. Mal-information is grounded on reality but either taken completely out of context or manipulated, with malicious intent to inflict harm on a person, organization or country.

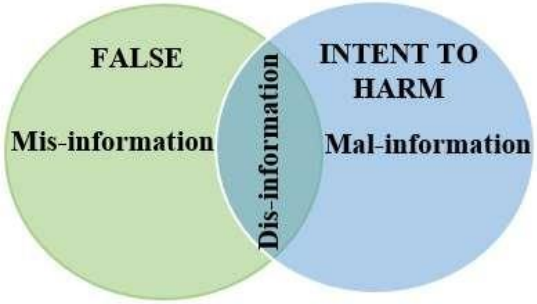


Fig.1. The 'information disorders' in social media

APPENDIX 2: LIST OF PUBLICATIONS (ACCEPTED)

Rumour Detection in Benchmark Dataset using Attention-Based Residual Networks

Akshi Kumar[1], Akshat Shrivastava[2]

[1,2]*Department of Computer Science & Engineering, Delhi Technological University, Delhi, India*

[1]*akshikumar@dce.ac.in*, [2]*akshatshrivastava2108@gmail.com*

Abstract

With the meteoric advancements in social media, a variety of information has become readily accessible to the public. Social media has become paramount and it influences people significantly through information. The sheer volume of information diffusion has led to an imperative need for questioning the tangibility of information. Rumors are an imperious threat to the credibility of the information sources. Rumors and non-rumors have to be meticulously separated from each other so that only the verified information reaches the public. This makes it necessary to look into the development of such tools or models that can detect rumors at an early stage and help in curbing their spread. In this paper, we have proffered an Attention-based Residual Network (ARN) model for rumour detection which employs residual blocks having skip connections, in combination with an attention mechanism. An early fusion strategy is used to combine the context-based (text + meta-features) and user-based features before feeding the combination to the ARN model, which outputs the final label as rumour or non-rumour. We have evaluated our proposed model on the PHEME dataset and the results validate superior classification performance to the state-of-the-art.

Keywords: *Rumour; Deep learning; Attention; Residual network; Early fusion; Text classification*

1. Introduction

In today's world, social media has taken over the conventional methods of communication and has brought about a paradigm shift in the way information is conveyed to a large audience. The time required for a piece of information to go viral across the world has reduced exponentially. The same piece of news or information may be reported by thousands or millions of people around the globe. There is a high variation in the information procured from these different sources. These variations lead us to believe that most of this information must have come from unverified sources. A surfeit of unverified information is disseminated on social platforms on a daily basis. A statement is considered as a rumor if its current status is unverified, irrespective of it being true or false. A rumour in circulation that later turns out to be false can be classified into one of the three categories: mis-information, dis-information and mal-information. They can be segregated from one another on the basis of intent of creation and proportion of truth. Mis-information is the result of an error and is not produced deliberately to cause harm. Dis-information is a work of pure fiction and consists of blatant lies created wantonly to cause harm. On the other hand, mal-information is based on real-life events but it is purposefully taken out of context or misrepresented with a vitriolic intent to exact harm.

REFERENCES

[1] A.K. Tripathi, K. Sharma, M. Bala, A. Kumar, V.G. Menon, A.K. Bashir. A Parallel Military Dog based Algorithm for Clustering Big data in Cognitive Industrial Internet of Things. IEEE Transactions on Industrial Informatics. 2020; 1–1. doi: 10.1109/TII.2020.2995680.

[2] V.G. Menon, M.R. Khosravi. Preventing hijacked research papers in fake (rogue) journals through social media and databases. Library Hi Tech News. 2019 Jun 17.

[3] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter. Detection and resolution of rumours in social media: A survey. ACM Computing Surveys (CSUR). 2018 Feb 20; 51(2):1-36.

[4] A. Kumar, S.R. Sangwan. Rumor Detection Using Machine Learning Techniques on Social Media. International Conference on Innovative Computing and Communications 2019 (pp. 213-221). Springer, Singapore.

[5] F.F. Ting, K.S. Sim. Self-regulated multilayer perceptron neural network for breast cancer classification. In2017 International Conference on Robotics, Automation and Sciences (ICORAS) 2017 Nov 27 (pp. 1-5). IEEE.

[6] A. Zubiaga, M. Liakata, R. Procter. Exploiting context for rumour detection in social media. In International Conference on Social Informatics 2017 Sep 13 (pp. 109-123). Springer, Cham.

[7] L. Bounegru, J. Gray, T. Venturini, M. Mauri. A Field Guide to 'Fake News' and Other Information Disorders. A Field Guide to" Fake News" and Other Information Disorders: A Collection of Recipes for Those Who Love to Cook with Digital Methods, Public Data Lab, Amsterdam (2018). 2018.

[8] J. Cao, J. Guo, X. Li, Z. Jin, H. Guo, J. Li. Automatic rumor detection on microblogs: A survey. arXiv preprint arXiv:1807.03505. 2018 Jul 10.

[9] T. Takahashi, N. Igata. Rumor detection on twitter. In The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems 2012 Nov 20 (pp. 452-457). IEEE.

[10] F. Yang, Y. Liu, X. Yu, M. Yang. Automatic detection of rumor on Sina Weibo. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics 2012 Aug 12 (pp. 1-7).

[11] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, S. Shah. Real-time rumor debunking on twitter. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management 2015 Oct 17 (pp. 1867- 1870).

[12] Y. Liu, S. Xu. Detecting rumors through modeling information propagation networks in a social media environment. IEEE Transactions on computational social systems. 2016 Oct 10; 3(2):46-62.

[13] S. Wang, T. Terano. Detecting rumor patterns in streaming social media. In2015 IEEE International Conference on Big Data (Big Data) 2015 Oct 29 (pp. 2709-2715). IEEE.

[14] Z. Zhao, P. Resnick, Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In Proceedings of the 24th international conference on World Wide Web 2015 May 18 (pp. 1395-1405).

[15] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K.F. Wong, M. Cha. Detecting rumors from microblogs with recurrent neural networks.

[16] T. Chen, X. Li, H. Yin, J. Zhang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining 2018 Jun 3 (pp. 40-52). Springer, Cham.

[17] Z. Jin, J. Cao, H. Guo, Y. Zhang, J. Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In Proceedings of the 25th ACM international conference on Multimedia 2017 Oct 19 (pp. 795-816).

[18] T.N. Nguyen, C. Li, C. Niederée. On early-stage debunking rumors on twitter: Leveraging the wisdom of weak learners. InInternational Conference on Social Informatics 2017 Sep 13 (pp. 141-158). Springer, Cham.

[19] A. Zubiaga, G. Wong Sak Hoi, M. Liakata, R. Procter. PHEME dataset of rumours and non-rumours [Internet]. figshare; 2016 [cited 2020 May 24]. Available from: https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619/1

[20] S.A. Alkhodair, S.H. Ding, B.C. Fung, J. Liu. Detecting breaking news rumors of emerging topics in social media. Information Processing & Management. 2020 Mar 1; 57(2):102018.

[21] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365. 2018 Feb 15.

[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation."(2014). arXiv:1406.1078

[23] A. Kumar, A. Shrivastava. (in press). “Rumour Detection in Benchmark Dataset using Attention-Based Residual Networks.” International Journal of Advanced Science and Technology. (2020).