SOFT COMPUTING FOR RUMOUR ANALYTICS ON BENCHMARK TWITTER DATA

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
SOFTWARE ENGINEERING

Submitted by:

Harshita Sharma
2K17/SWE/09

Under the Supervision of

Dr. AKSHI KUMAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2019

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **HARSHITA SHARMA**, 2K17/SWE/09 student of M.Tech Software Engineering, hereby declare that the project Dissertation titled "**Soft Computing For Rumour Analytics On Benchmark Twitter Data**" which is submitted by me to the Department of Computer Science, **Delhi Technological University**, Delhi in partial fulfilment of the requirement for the award of the 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: Delhi HARSHITA SHARMA

Date: June, 2019

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project Dissertation titled "Soft Computing For Rumour Analytics On Benchmark Twitter Data" which is submitted by Harshita Sharma, 2K17/SWE/09, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the Degree of Master of Technology, is a record of the 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: June, 2019 SUPERVISOR

Assistant Professor

Dept. of Computer Science and Engineering,

Delhi Technological University

ABSTRACT

As social media is a fertile ground for origin and spread of rumours, it is imperative to detect and deter rumours. Various computational models that encompass elements of learning have been studied on benchmark datasets for rumour resolution with four individual tasks, namely rumour detection, tracking, stance and veracity classification. Quick rumour detection during initial propagation phase is desirable for subsequent veracity and stance assessment. This research presents the use of adaptive and heuristic optimization to select a near-optimal set of input variables that would minimize variance and maximize generalizability of the learning model, which is highly desirable to achieve high rumour prediction accuracy. An empirical evaluation of hybrid filter-wrapper on PHEME rumour dataset is done. The features are extracted initially using the conventional term frequency-inverse document frequency (TF-IDF) statistical measure and to select an optimal feature subset two filter methods, namely, information gain and chi-square are separately combined with three swarm intelligence-based wrapper methods, cuckoo search, bat algorithm and ant colony optimization algorithm. The performance results for the combinations have been evaluated by training three classifiers (Naïve Bayes, Random Forest and J48 decision tree) and an average accuracy gain of approximately 7% is observed using hybrid filter-wrapper feature selection approaches. Chi-square filter with Cuckoo and ACO give the same maximum accuracy of 61.19% whereas Chi-square with bat gives the maximum feature reduction selecting only 17.6%

features. The model clearly maximizes the relevance and minimizes the redundancy in feature set to build an efficient rumour detection model for social data.

Due to the ever increasing use and dependence of netizens on social media, it has become a fertile ground for breeding Rumours. This work aims to propose a model for Potential Rumour Origin Detection (PROD) to enable detection of users who can be likely rumour originators. It can not only help to find the original culprit who started a rumour but can aid in veracity classification task of the rumour pipeline as well. This work uses features of the user's account and tweet to extract meta-data. This meta-data is encoded in an 8-tuple feature vector. A credibility quotient for each user is calculated by assigning weights to each parameter. The higher the credibility of a user, less likely it is to be a rumour originator. Based on the credibility, a label is assigned to each user indicating whether it can be a potential rumour source or not. Three supervised machine learning algorithms have been used for training and evaluation and compared to a baseline zeroR classifier. The results have been evaluated on benchmark PHEME dataset and it is observed that the multi-layer perceptron classifier achieves the highest performance accuracy, that is, an average 97.26% for all five events of PHEME to detect potential rumour source.

ACKNOWLEDGEMENT

I would like to express my immense gratitude to the Almighty God who guided me from within and without and without whom it was impossible. I am very thankful to Dr. Akshi Kumar (Asst. Professor, Computer Science Eng. Dept.) She provided me with immense support and guidance for the project. I would also like to express my gratitude to my family without whom it would have been impossible.

I am also grateful to all the faculty members of the Computer Science Engineering Dept. of Delhi Technological University, all scholars of our Web Research Group (WRG) and the university for providing me with the laboratories, infrastructure, testing facilities and environment which allowed me to work without any obstructions. I would also like to appreciate the support provided to me by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

HARSHITA SHARMA

(2k17/SWE/09)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

TF-IDF: Term frequency – document frequency

BOW: Bag of Words

IG: Information Gain

CS: Chi-square

ACO: Ant Colony Optimization

A_c: Accuracy

P_r: Precision

R_e: Recall

F_{meas}: F-measure

NB: Naïve Bayes

SVM: Support Vector Machines

DT: Decision Trees

MLP: Multilayer Perceptron

CHAPTER 1

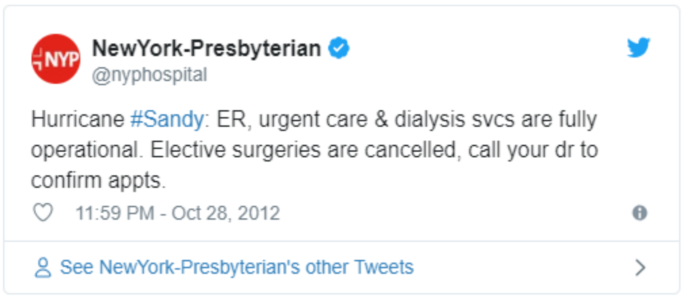
INTRODUCTION AND OVERVIEW

This chapter presents a concise and comprehensive introduction to the research problem addressed in the thesis work followed by a brief overview of the techniques used. The chapter ends with an outline of the rest of the thesis.

1.1 Introduction

Social media is an indispensable and ubiquitous part of life today. It has enabled mobilization of information eliminating communication and demographical barriers. With the advent of smartphones, social media has become all the more easily accessible to people from all walks of life. Not only is it a subtle way to stay in touch with people we know, it is a platform to share anything and everything from our photos and visits to our moods. Be it political issues like elections, natural disasters like floods and earthquakes, terrorist attacks, epidemics, issues relating to celebrities, the public in general, or any socio-cultural issue, social media users do not shy away from expressing their opinions and posting updates.

For example, when hurricane Sandy hit USA's East coast, power went out thus cutting off most traditional forms of communication like television and radio, people took to social media to stay updated. A hospital in New York used took to social media to inform people about how they can stay safe. An instance is shown in fig 1.1.


NewYork-Presbyterian
@nyphospital

Hurricane #Sandy: ER, urgent care & dialysis svcs are fully operational. Elective surgeries are cancelled, call your dr to confirm appts.

11:59 PM - Oct 28, 2012

See NewYork-Presbyterian's other Tweets

Fig 1.1. Use of Social Media during Emergency Situations

A more recent example is that of Kerala Floods in India's southern region where not just the government and disaster relief organizations but also ordinary citizens took to social media for both rescue and relief operations. People used twitter, WhatsApp and whatever they could to provide people with information about helpline numbers, free medical and other services. Social media has thus become an important source for not only gathering opinions but also for disseminating information. Governments, companies and others look to social media for feedback on new schemes and products. Undoubtedly this pervasive web has enriched our lives but there is a flip side to the phenomenon as well. Cyber bullying [1] [2], psychological distress [3] [4], echo chambers [5], viral misinformation [6] and cascading rumours [7] are key social problems associated with the evolving web. Social media is equally easily accessible to those who want to help as to those who want to create trouble. Given how easy it is for anyone to create a fake account, rumor mongers can take great advantage of the increasing reliance on social media for staying updated about current events. A recent study by researchers at MIT [8] show that fake and rumorous posts spread much faster than authentic ones. People keep on repeatedly posting and sharing a piece of information which thus proliferates across entire networks and affects millions. Though the updates can help government, disaster relief organizations and the general public during critical situations but the falsification or fabrication of information may harm the status, privacy and lives of people.

1.2 Defining Rumours

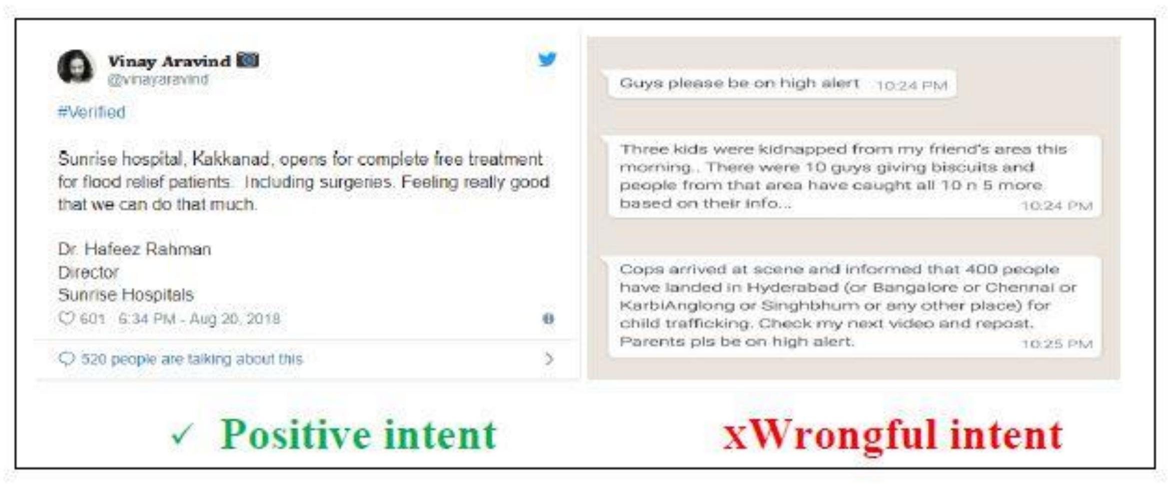
Simply put, a rumor is an assertion whose truth value is unverified. Rumours have been studied and analysed from a range of perspectives, and within and across different disciplines [9]. Most of the definitions given in the literature agree with that of major dictionaries such as the Oxford English Dictionary, which defines a rumour as “a currently circulating story or report of uncertain or doubtful truth”, as well as the Merriam-Webster dictionary defining it as “information or a story that is passed from person to person but has not been proven to be true”. Irrespective of the underlying story being ultimately proven true or false, or remaining unsubstantiated, a rumour circulates while it is yet to be verified. A number of researchers have extended the definition of rumour. For instance, DiFonzo and Bordia, 2007 [10] define rumours as “unverified and instrumentally relevant information statements in circulation that arise in contexts of ambiguity, danger or potential threat, and that function to help people make sense and manage risk”. Moreover, Allport and Postman, 1946 [11] posit that one of the main reasons why rumours circulate is that “the topic has importance for the individual who hears and spreads the story”. The authors also emphasise that “newsworthy events are likely to breed rumors” and that “the amount of rumor in circulation will vary with the importance of the subject to the individuals involved times the ambiguity of the evidence pertaining to the topic at issue” [12].

Consistent with these definitions, Zubiaga et al., 2016 [13] have adapted a definition to the context of breaking news: a rumour is a “circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient skepticism and/or anxiety so as to motivate finding out the actual truth”. In the context of journalism, spreading rumours can have harmful consequences for the reputation of a news organisation if they are used in reporting and later proven false, and hence being able with confidence to quickly assess whether information has not yet been verified as breaking news unfolds is crucial.

1.3 Motivation and Scope

Given how easy it is for anyone to create a fake account, rumor mongers can take great advantage of the increasing reliance on social media for staying updated about current events. It is not easy for the general public to identify a rumor from non-rumor as they have neither the means nor the will to verify the credibility of each post they come across. Rumors spread like wild fire and have in the past created much trouble. Not only are they misleading but are potentially harmful to the status, privacy and even the lives of people. The increasing over-reliance on social media as a source of information, has also given rise to the trend of posting fake news and posts thus sparking rumors. There is usually no means to check the authenticity of social media posts and even when there is people are so hasty and eager to share it, most of them do not bother to verify what they are posting. This makes social media websites a fertile ground for breeding rumors.

All kinds of information are posted on social media. Some are good, some bad, some useful while some useless. Not all information is rumorous. The following figure 1.2 depicts the example use of social media to broadcast information with positive and wrongful intent.



Vinay Aravind
@vinayaravind

#Verified

Sunrise hospital, Kakkanad, opens for complete free treatment for flood relief patients. Including surgeries. Feeling really good that we can do that much.

Dr. Hafeez Rahman
Director
Sunrise Hospitals

601 6:34 PM - Aug 20, 2018

520 people are talking about this

Guys please be on high alert 10:24 PM

Three kids were kidnapped from my friend's area this morning.. There were 10 guys giving biscuits and people from that area have caught all 10 n 5 more based on their info.. 10:24 PM

Cops arrived at scene and informed that 400 people have landed in Hyderabad (or Bangalore or Chennai or KarbiAnglong or Singhbhum or any other place) for child trafficking. Check my next video and repost. Parents pls be on high alert. 10:25 PM

Fig 1.2. Social Media and its use with positive and wrongful intent.

Thus, given the tremendous power that social media has to influence people, it is imperative that posts on social media be monitored, potential rumors be identified and further be tracked and mitigated before the post becomes viral and affects millions of people.

1.4 The Rumour Analytics Pipeline

The rumor analytics process consists of four phases or subtasks as follows:

- **Rumor Detection**: In this step, a binary classifier is given as input a stream of posts and it outputs each post labelled as being a rumor or non-rumor. It is very important when working with emerging rumors.

- **Rumor Tracking**: Given a rumor as input, either in the form of a descriptive sentence or in the form of keywords, the social media is monitored for relevant posts describing the rumor. These related posts are then given as output.

- **Stance Classification**: This component determines the orientation of each related post, output by the rumor tracking component and outputs posts labelled by a stance like supporting, denying or querying a rumor.

- **Veracity Classification:** This component outputs the veracity of a rumorous post by using the outputs of the previous two components as inputs as well as other online sources like news websites optionally. The output can be either only a truth value or additionally some context like links to online data sources.

The detection of **rumour origin** is also a subtask that tracks the original or the first user who posted that content. Fig 1.3 depicts a graphical representation of the rumour analytics pipeline.

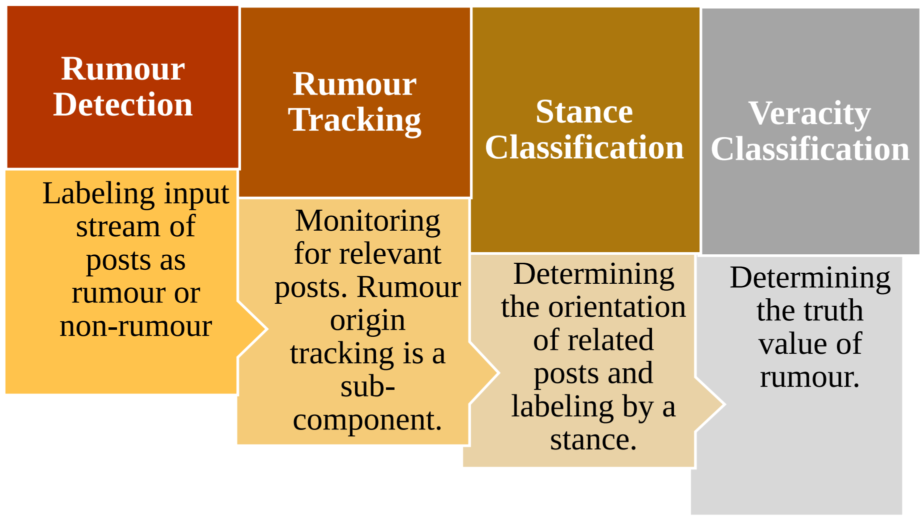

Rumour Detection
Labeling input stream of posts as rumour or non-rumour
Rumour Tracking
Monitoring for relevant posts. Rumour origin tracking is a sub-component.
Stance Classification
Determining the orientation of related posts and labeling by a stance.
Veracity Classification
Determining the truth value of rumour.

Fig 1.3.The Rumour Analytics Pipeline

1.5 Soft Computing

Soft computing is a computing approach that is modeled on the characteristics of the human intelligence. Unlike hard computing, it has tolerance for and works well with inexact, imprecise, uncertain and partial results. It can use approximation to achieve robust, tractable solutions to computationally hard and NP-complete problems. It encompasses machine learning, neural networks, fuzzy logic, evolutionary computing and probabilistic reasoning. Fig 1.4 illustrates a broad classification of various soft computing techniques.

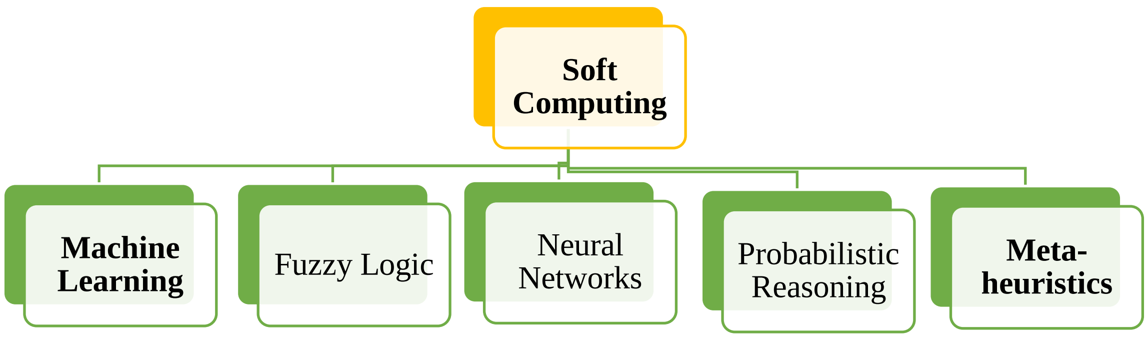


Fig 1.4. Soft Computing Techniques

1.5.1 Machine Learning and Optimization

Machine Learning is a branch of Artificial Intelligence. It provides the ability to computers to automatically learn from experience without being explicitly programmed. The input can be data samples, instructions or direct experience. The output can be decision based like in classification tasks, certain patterns inferred by the computer like in pattern recognition tasks, etc. The aim is to let the system adjust itself by tuning the learning parameters based on its performance.

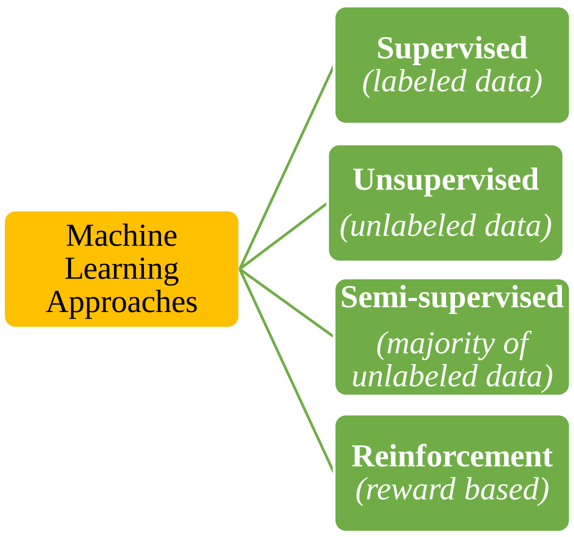


Fig 1.5. Machine Learning Approaches

The classification of machine learning algorithms is illustrated in fig 1.5 and these are explained below.

- **Supervised Machine Learning:** It is a class of machine learning algorithms that learn by comparing their output response with the actual response provided to them. Thus, they use labelled data as input and try to adjust their parameters to get their response as close to the target response as possible. Once trained, these algorithms are presented with an unlabeled data set which they have to label. Supervised ML algorithms can be used for both classification which predicts a discrete value or category that the data belongs to, as well as for regression tasks which predicts a numerical or real, continuous value. Sentiment analysis and spam filtering are examples of classification tasks while stock price prediction and time-series forecasting are examples of a regression task. While some algorithms like Support Vector Machines and Decision trees can be used for either of the two tasks, others like Naïve Bayes and Logistic Regression are suitable for only classification and regression respectively.

- **Unsupervised Machine Learning:** These are a class of machine learning algorithms that learn from an unlabeled data set, i.e., the algorithm has no target output to compare its response with. They group data together based on recognition of certain patterns and commonalities. Data are grouped together based on the absence or presence of these commonalities amongst them. Unsupervised ML algorithms are classified as clustering or associating algorithms. Clustering algorithms like the k-means group data together based on insights and inherent patterns among groups. Practical applications of clustering include pattern recognition, statistical data analysis and information retrieval. Association rule learning algorithms like the Apriori algorithm, on the other hand, discover relations among variables in the data set that associate variables in one set to the other. Practical applications include web usage mining and market based analysis.

- **Semi-supervised Machine Learning:** This class of machine learning algorithms takes as input semi-labeled data with majority of the data being unlabeled. The goal being to classify the unlabeled data deducing patterns and learning from the labeled data. Semi-supervised has a lot of practical applications where a lot of data is available and it is cumbersome and time intensive to label each set of features. Examples include speech analysis, protein sequence classification and web content classification. Self-training, generative models, semi-supervised support vector machines (S3VMs) are examples of semi-supervised learning algorithms.

- **Reinforcement Machine Learning:** It is a reward based machine learning approach, wherein the closer the output is to the output higher is the reward. Unlike supervised learning, the algorithm is not provided the actual target outputs, instead it learns from experience according to the reward which can be either positive or negative. Industrial automation is an application are of reinforcement learning. Q-learning and Sate-

Action-Reward-State-Action (SARSA) are examples of some reinforcement learning algorithms.

Given the huge amounts of datasets, a large number of features need to be extracted and analyzed resulting in large feature matrices. Not all of these features are relevant to the classification task. Moreover, the larger the feature set, more will be the time required for training the model. It is desirable to use only the most important features for learning. This selection of best or optimal features is done using optimization algorithms.

The term optimization refers to making the best and most efficient use of a given situation. An optimization problem, thus, is one where we have to find the best solution among many feasible ones. A feasible solution is one that satisfies all the constraints of the given problem. The best solution, on the other hand, is one that maximizes or minimizes certain problem parameters. Although the terms best and efficient can refer to various parameters depending on the problem at hand, in most real word applications we either need to minimize the cost, time or some other limited resource, or maximize the efficiency of the system.

Optimization algorithms can broadly be classified into deterministic and stochastic algorithms. In deterministic models, the output is completely determined from the initial conditions and parameters. Stochastic models, on the other hand have an inherent randomness such that the same initial conditions and parameters will lead to an ensemble of different outputs.Heuristic algorithms are the class of stochastic optimization algorithms that aim to find close to optimal solutions to NP-Hard optimization problems. The solutions may not be the best but they are reasonably acceptable. Meta-heuristics go beyond heuristics and are a level up and generally give better results than simple heuristics. Almost all meta-heuristics use a trade-off of randomization and local search. Fig 1.6 presents a classification of optimization algorithms.

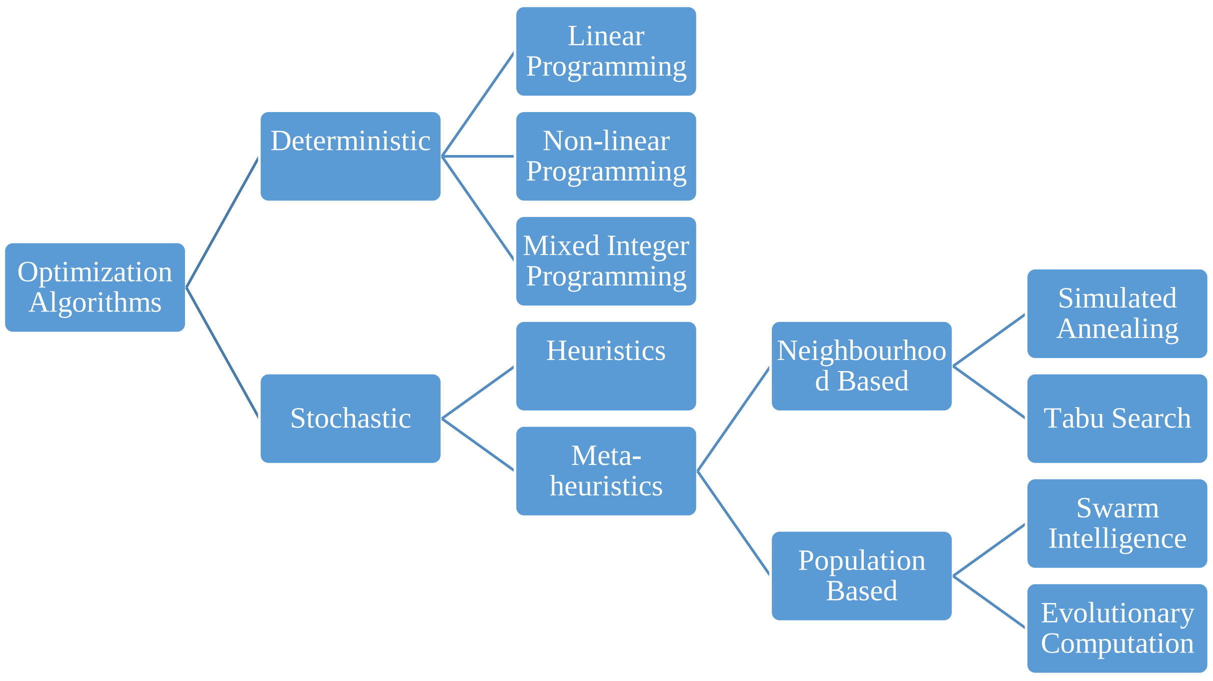


Fig 1.6. Classification of Optimization Algorithms

1.5.2 Swarm Intelligence

Swarm intelligence algorithms are a class of population-based nature inspired metaheuristics that take inspiration from birds, animals and other natural phenomenon. Swarm algorithms usually work over a population of agents, also called as hosts. Swarm techniques make use of collection, decentralization, distribution and self-organization of agents to arrive at an optimal solution. The most popular among the swarm-based algorithms are those inspired by the behaviour of species in nature like birds, ants, insects, etc. Some examples of popular swarm algorithms are particle swarm optimization, ant colony optimization, cuckoo search, etc.

1.6 Research Objectives

This research proffers a hybrid filter-wrapper rumour detection model to identify new rumours in a timely manner. The objective to use a hybrid is to maximize the relevance and minimize the redundancy in feature set which is used to train the learning model. The hybrid takes the advantage of both the efficiency of filters and the accuracy of wrappers [14]. The hybrid filter-wrapper generates an optimal feature matrix to train the learning model. To create the initial feature matrix, term frequency-inverse document frequency (TF-IDF) [15] is used. While most researchers working in this research domain focus on the social, temporal, propagation and network-based features [16] [17] [18] [19] [20], this research focuses exclusively on the textual content of the tweet. Linguistic features are abundant and are significant in the early phase of propagation. Moreover, given the restriction of microblogging sites regarding the length of tweets, users try to be concise, use abbreviations and keywords. All these can assist in identifying rumours.

The subtask of rumour origin of the rumour analytics pipeline has been explored the least and this work aims to detect a user who can be a potential rumour origin based on certain user account and tweet content based features. The work proposes a PROD (Potential Rumour Origin Detection) Model which labels each user based on whether he can be a potential rumour source or not defining a credibility quotient derived from the values of 8-tuple feature vector extracted based on user data and tweet content. Finally, 3 supervised ML algorithms Naïve Bayes (NB), Support Vector Machines (SVM) and Multilayer Perceptron (MLP) are used to evaluate the performance of the model. Thus, the key questions addressed in the work include:

- ❖ How to detect rumours on social media?
- ❖ Can feature selection techniques like filter and wrapper methods be used to detect rumours?
- ❖ Which features can be used to identify a user as being a potential rumour source?

Thus, the primary research objectives of the work can be stated as follows:

Research Objective I: To build a rumour detection model with hybrid filter-wrapper method

Research Objective II: To evaluate the performance of various feature selection methods for rumour detection on benchmark PHEME dataset.

Research Objective III: To identify the features which are indicative of a user being a potential rumour source.

1.7 Organization of the Thesis

The rest of the thesis is organized as follows: chapter 2 presents a literature survey in the field of rumour analytics focusing on the subtasks of rumour detection and rumour origin. Chapters 3 and 4 elucidate the models proposed for rumour detection and rumour origin respectively. Chapter 5 presents and discusses the results obtained from training using the models proposed. Chapter 6 concludes the thesis and presents future research directions.

CHAPTER 2

LITERATURE REVIEW

This chapter briefly reviews the work done in various subtasks of the rumour analytics field and extensively reviews the most notable research in the field of rumour detection and origin.

2.1. Social Media and Rumours

Social media has been explored by researchers for sentiment analysis [21] [22] [23] [24], sarcasm detection [25], cyberbullying detection [1] [2], expert mining [26] and rumour analytics [7] [27] [6].

Rumor analytics is an active area of interest and research [28] [29] [30] [31] [32] [13] [33]. A number of tools have been built to analyse previously identified rumors [30] [34]. Some prominent work has been done in detecting known rumors by [35] [36] [37]. This is useful for long standing rumors and can be helpful in tracking a rumor once it has been identified. There has also been work on related tasks like stance classification [38] [39] [35] [40] which classifies posts discussing a rumor as in support, denial or questioning a rumor. A supervised classifier is trained given a labelled dataset consisting of tweets and the stance for the tweets is predicted. However, only rumors are dealt with, non-rumors are not handled in their work. It is assumed that the input is already processed and cleaned and thus consists of only rumors.

Work has also been done on veracity classification [38] [41] [42] [43] [44] [45] [46] [47]. The tasks of stance and veracity classification are complementary to the task of detection; as rumours detected by the rumour detection sub-system system can be provided as input to a classifier determining stance in the input rumours and/or their veracity.

However, this previous step of distinguishing between out rumours and non-rumours is largely unexplored, and most work deals directly with subsequent steps. Researchers have explored various types of features of the tweets broadly categorized as content based (like POS, NER etc.), pragmatic features (like emoticons, event etc) and network specific features (like hashtags, urls, retweet etc.), user features (like friends count, followers count, etc.), client based and propagation-based features. Swarm algorithms have not been explored much in the context of rumor analytics. To the best of our knowledge, only a few works have used swarm intelligence for rumor analytics [48] [49].

2.2 Related Work in Rumour Detection

The majority of research in rumour analytics has been carried out in the area of rumour veracity classification. The work presented in this paper, however, is based on the primary task of rumour detection. Table 2.1 summarizes the most notable work. Most researchers have worked on microblogging sites, namely, Twitter or Sina Weibo.

Table 2.1. Related Work in Rumour Detection

Authors	**Microblogging site used**	**Features used**	**ML algorithms used**
Yang et al. [16]	Sina Weibo	Content based Account based Propagation based Client based Location based	SVM with RBF kernel
Zhang et al. [17]	Sina Weibo	Shallow features Content based features User based implicit features	SVM
Jin et al. [18]	Sina Weibo	User, content, social, semantic, structure and propagation based features	SVM
Wang and Terano[19]	Twitter	Textual features, temporal features, propagation structures and behaviour of users	Proposed a pattern matching algorithm for rumour pattern detection
Yang et al. [20]	Sina Weibo	Content based features, micro-blogging features, account based features, topological features of the network	NB
Sahana et al. [50]	Twitter	Tweet content features User account features	J48

Wu et al. [41]	Sina Weibo	Propagation structures Semantic features	Hybrid SVM using graph kernel and radial basis function.
Hamidian and Diab [36]	Twitter	Network and Twitter specific features Meta linguistic features Pragmatic features	J48
Zhao et al. [51]	Twitter	Statistical tweet features	Clustering, SVM, DT
Castillo et al. [52]	Twitter	Message, User, Topic and Propagation features	J48
Kwon et al. [53]		Temporal, structural, and linguistic features	DT, Random Forest, SVM
Ma et al. [54]	Sina Weibo	Textual features	NB, Logic, SMO, MLP, K-NN, Random Forest, Random Tree
Liu at al. [38]	Twitter	Language, network propagation, account, user and meta features	SVM, DT, Random Forest
Gupta et al. [55]	Twitter	User, Network, event, Content based features	SVM, NB, K-NN, J48
Jin et al. [46]	Sina Weibo	Content, user and social relations based features	SVM
Ma et al. [44]	Twitter, Sina Weibo	Content based, user based and propagation based features	DT, Random Forest, SVM

Giasemidis et al. [56]	Twitter	Message based, user based, network based, time series based, propagation based features	Logistic Regression, Linear SVM, RBF based SVM, DT, Random Forests, NB, Neural Networks
Kwon et al. [57]	Twitter	User, structural, linguistic, and temporal features	Random Forest
Jin et al. [58]	Sina Weibo	Image based features, textual, user, propagation based features	SVM, Logistic Regression, KStar, Random Forest
Zhang et al. [59]	http://www.liuyanbaike.com/ - preeminent online resource for verifying and debunking rumours in China	Textual, content based, multimedia related features	Logistic Regression
Ma et al. [43]	Twitter, Sina Weibo	Temporal, content, user and propagation based features	SVM, DT, Random Forest, RNN, LSTM, GRU
Chen et al. [60]	Twitter, Sina Weibo	Textual, temporal features	DT, SVM, LK-RBF, ML-GRU, LSTM, RNN
Ruchansky et al. [61]	Twitter, Sina Weibo	Temporal, textual, user features	DT, SVM, GRU, LSTM

Jin et al. [62]	Twitter, Sina Weibo	Textual, visual, social contextual features	Logistic classifier, RNN, LSTM
Yu et al. [63]	Twitter, Sina Weibo	Ensemble features	SVM, DT, GRU, CNN, Random Forest
Nguyen et al. [64]	Twitter	Ensemble, Twitter and Epidemiological features	1-layer GRU-RNN, 1-layer LSTM, CNN+LSTM, 2-layer GRU-RNN, Tanh RNN

DT: Decision Tree; SVM: Support Vector Machine; NB: Naïve Bayes; RBF: Radial Basis Function; SMO: Sequential minimal optimization; MLP: Multilayer Perceptron; K-NN: K nearest neighbours; RNN: Recurrent Neural Networks; LSTM: Long short-term memory; GRU: Gated Recurrent Units; CNN: Convolutional Neural Networks.

Rumour detection is commonly referred to as a binary classification task with features extracted from textual comments, user profiles and information propagation patterns. Among these feature types, the text-based rumour detection techniques have been prominently studied because of the content stability and efficiency of textual feature extraction is higher as compared to modelling a user profile to detect a candidate rumour post. For analysing textual features, researchers have used TF-IDF, entropy ratio and LDA (Latent Dirichlet Allocation) [52] [58] [51] [60]. But none of the studies, to the best of our knowledge, have discussed a filter-wrapper method to train a model of rumour detection using textual features of the tweet. The next section expounds the details of the proposed hybrid filter-wrapper model.

2.3 Rumour Origin and Related Work

As important it is to detect and debunk a rumour, equally important is finding the origin of the rumour post. Doing so, will not only help catch the culprit, but can hint screening agencies to pro-actively keep a check on users activities. This subtask of rumour analytics has been explored the least and this work aims to detect a user who can be a potential rumour origin based on certain user account and tweet content based features. The work proposes a PROD (Potential Rumour Origin Detection) Model which labels each user based on whether he can be a potential rumour source or not defining a credibility quotient derived from the values of 8-tuple feature vector extracted based on user data and tweet content. Finally, 3 ML algorithms based on the supervised approach namely NB, SVM and MLP are used to evaluate the performance of the model.

The rest of the paper is ordered as follows: the second section summarizes the existing work, the third section expounds the proposed potential rumour origin detection (PROD) model, the fourth section presents the results. The last section concludes the work and presents some future directions.

The area of rumour origin detection has been explored little. We summarize some of the most notable works. The first systematic study of finding rumour sources used the SIR model where the authors built an estimator for rumour source in trees and gen-eral graphs [65]. They use the approach of maximum likelihood. The problem of rooting out a single rumour source from a set of suspect nodes under an SI model has also been studied [66]. They build a maximum a posteriori (MAP) to identify the rumour source. Some authors introduce monitors in the network and build algorithms based on the approach that which receive information from the network [34] [67]. Rumour sources are detected based on which nodes receive more information. In another work, researcherscompare time of tweet posting to find the origin [50]. A track of rumour propagation trails and empirical investigation of the rumour teller's position using statistics has also been done [68]. The problem of single rumour source detection with multiple observations has also been addressed [69]. The authors pro-pose a unified inference framework using the concept of union rumour centrality.

CHAPTER 3

PROPOSED MODEL FOR RUMOUR DETECTION

This chapter describes in detail the proposed model for the subtask of rumour detection based on hybrid filter and wrapper techniques.

3.1 Introduction

Rumour detection is a quintessential text classification task which intends to categorize the incoming social media post as rumourous or not. Feature engineering is a non-trivial sub-task in text classification, which includes feature extraction, feature transformation and/or feature selection. Feature extraction takes care of the generation of features from data that are in a format that is difficult to analyze directly or are not directly comparable (e.g. images, time-series, etc.). Feature transformation transforms the existing features in order to create new ones based on the old ones. On the other hand, feature selection, selects the features having maximum influence on the targeted variable. If the only intent is to achieve dimensionality reduction in an existing dataset, we can either use feature transformation or feature selection methods. But if the intent is to understand the physical interpretation of the features identified as "important" or if we are trying to limit the amount of data that needs to be collected for analytics, then only feature selection can work. Feature selection methods can further be classified into filters, wrappers, embedded and hybrid methods [70]. This research proffers a hybrid filter-wrapper rumour detection model to identify new rumours in a timely manner. The objective to use a hybrid is to maximize the relevance and minimize the redundancy in feature set which is used to train the learning model.

The hybrid takes the advantage of both the efficiency of filters and the accuracy of wrappers [14]. The hybrid filter-wrapper generates an optimal feature matrix to train the learning model. To create the initial feature matrix TF-IDF [15] is used. While most researchers working in this research domain focus on the social, temporal, propagation and network-based features [50] [51] [52] [53] [54], this research focuses exclusively on the textual content of the tweet. Moreover, given the restriction of microblogging sites regarding the length of tweets, users try to be concise, use abbreviations and keywords. All these can assist in identifying rumours.

3.2 The Proposed Filter-Wrapper Rumour Detection Model

The rumour detection model uses a hybrid filter-wrapper method for rumour detection on benchmark PHEME dataset. It evaluates the importance of textual content of the source tweet by using TF-IDF as a feature extraction method. Combinations of 2 filters (information gain, chi-square) and 3 swarm intelligence-based wrapper methods (cuckoo search, bat algorithm, ant colony optimization) are assessed for feature selection. Three classifiers, namely, Naïve Bayes, Random Forest and J48 decision tree are trained using the optimal feature set generated by filter-wrapper to determine the performance of the proposed model in terms of accuracy, precision, recall and F-measure. The structural flow of the proposed model is shown in fig.3.1.

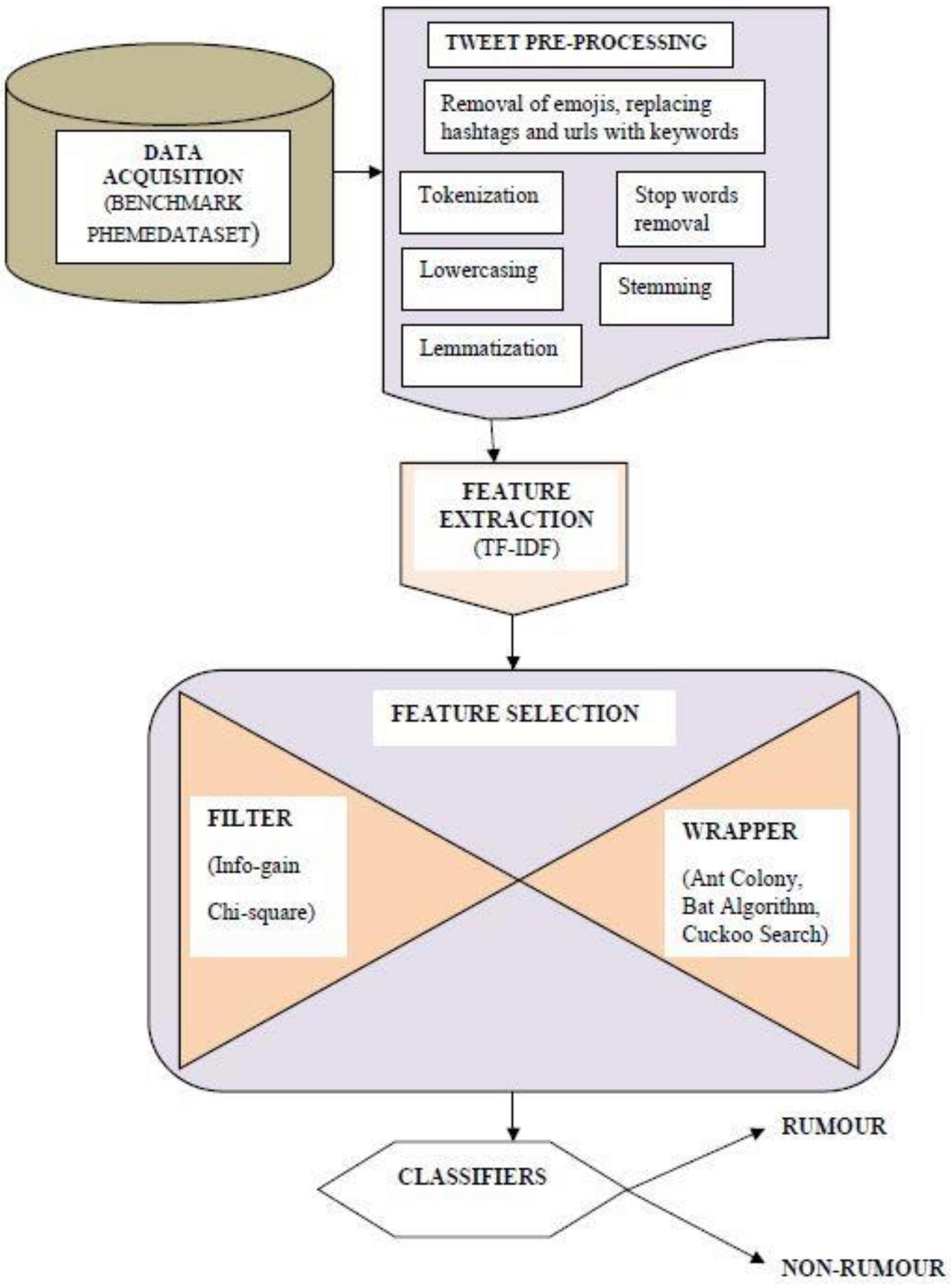

DATA ACQUISITION (BENCHMARK PHEMEDATASET)
TWEET PRE-PROCESSING
Removal of emojis, replacing hashtags and urls with keywords
Tokenization
Stop words removal
Lowercasing
Stemming
Lemmatization
FEATURE EXTRACTION (TF-IDF)
FEATURE SELECTION
FILTER (Info-gain Chi-square)
WRAPPER (Ant Colony, Bat Algorithm, Cuckoo Search)
CLASSIFIERS
RUMOUR
NON-RUMOUR

Fig 3.1. Systematic workflow of the proposed system

The following sub-sections expound the details.

3.2.1. Data Acquisition

The dataset used in this work is the PHEME dataset of rumours. It consists of labelled tweets for 5 breaking news events as follows:

- *#charliehebdo* - Around noon on 7th January 2015, two gunmen forced themselves into the offices of the French satirical weekly newspaper Charlie Hebdo in Paris and killed 12 people and wounded 11. The dataset contains 458 rumours and 1621 non-rumours.

- *#ferguson* - On 9th August, 2014, a white police officer, 28-year-old Darren Wilson fatally shot an African-American 18 year old named Michael Brown Jr, in Ferguson, Missouri. The officer reports an altercation between him and Brown when Brown attacked him. He later fed and was chased by Wilson. A total of twelve bullets were fired by Wilson, six of which hit Brown from the front. Several protests followed. The dataset contains 284 rumours and 859 non-rumours.

- *#germanwingscrash*– On 24th March, 2015, an Airbus A320-211 crashed 100 kilometres (62 mi) north-west of Nice in the French Alps . It was scheduled for the international Germanwings Flight 9525 from Barcelona–El Prat Airport in Spain to Düsseldorf Airport in Germany and killed all 144 passengers and six crew members. The crash was a deliberate one caused by the co-pilot diagnosed with suicidal tendencies and declared unfit for work by his doctor. The dataset contains 238 rumours and 231 non-rumours.

- *#ottawashooting* - October 22nd, 2014, a series of shooting took place Ottawa’s Parliament Hill. At the Canadian National War Memorial, Corporal Nathan Cirillo, a Canadian soldier on ceremonial sentry duty was fatally shot by Michael Zehaf-Bibeau. Zehaf-Bibeau then entered the nearby Centre Block parliament building, where members of the Parliament of Canada were attending caucuses. After wrestling with a constable at the entrance, Zehaf-Bibeau ran inside and had a shootout with parliament security personnel. He was shot 31 times by six officers and died at the scene. The dataset contains 470 rumours and 420 non-rumours.

• *#sydneysiege* - on 15-16th December, 2014 an armed gunman, Man Haron Monis held hostage ten customers and eight employees of a Lindt chocolate café in Sydney, Australia. There was a 16-hour standoff, two people were killed and a few injured.

3.2.2. Tweet Pre-processing

A piece of text includes a number of words some of which might be important while others may be unimportant. Text also includes punctuation marks, whitespaces and social media generated text might also include some special symbols, urls, emoticons and the like. Text must be pre-processed to remove useless words, whitespaces and other data that might act like noise. Pre-processing is an important step in text classification [71] [24] [72]. It includes a number of steps which are described below.

- *Removal of emojis and replacing urls and hashtags with keywords*

The first sub-task performed in the pre-processing step, was the removal of emojis. Then, the next sub task was to remove the urls. Since, the actual url was of little importance for our task, we replaced the actual links with the keyword ‘url’ to signify the presence of a link in a tweet. Similarly, the various hashtags in the tweets were replaced by the keyword ‘hashtag’, to acknowledge the presence of a hashtag in the source tweet. We used Python’s tweet pre-processor for this step.

- *Tokenization*

Tokenization or segmentation or lexical analysis involves splitting a longer piece of string into smaller chunks. The smallest meaningful chink is called a token and can refer to a word, a character, or even a sentence. We use word tokenization here and use NLTK library's word_tokenize() function.

- *Stop Words Removal*

Conjunctions, articles, determinants, propositions are examples of words that occur frequently but are essentially meaningless. These are referred to as stop words and are removed using NLTK's nltkcorpus' stopwordsas they do not contribute in discriminating a text.

- *Lowercasing*

Capitalization is removed from the sentence by converting all letters to lowercase.

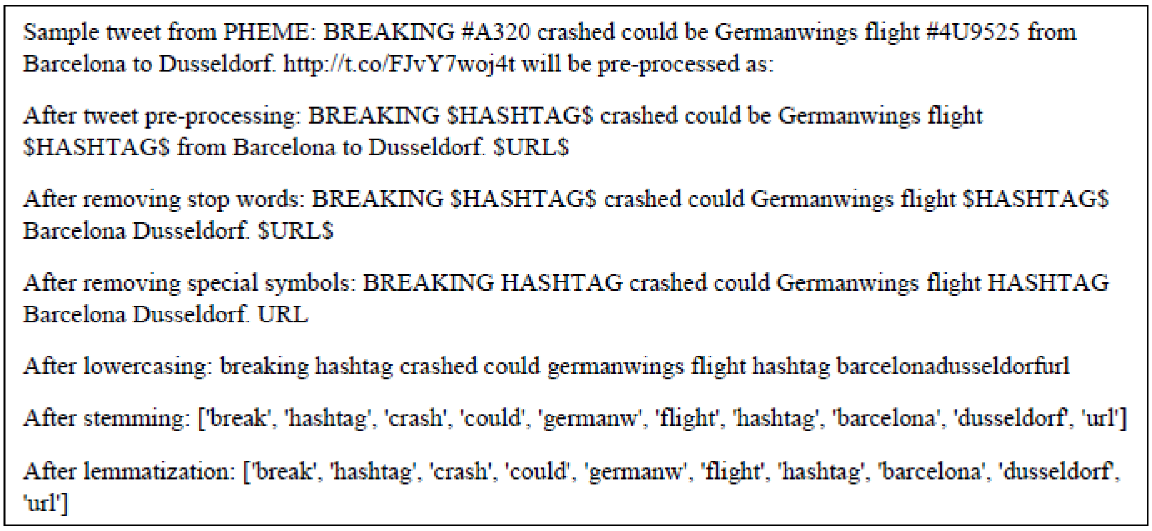
- *Stemming*

It is the process of reducing words in a text to their root form by considering only prefixes and suffixes. For example, words like talking, talked, talkative are all reduced to their root word 'talk'.TheNLTK'sstem's PorterStemmer is used.

- *Lemmatization*

It is related to stemming but is used to recognize canonical forms based on a word's lemma using NLTK'sWordNetLemmatizer. Lemmatization considers the morphology of the words using detailed corpus that links a word to its original lemma.

We illustrate the pre-processing step in detail for a sample source tweet from the PHEME dataset in fig.3.2.



Sample tweet from PHEME: BREAKING #A320 crashed could be Germanwings flight #4U9525 from Barcelona to Dusseldorf. http://t.co/FJvY7woj4t will be pre-processed as:

After tweet pre-processing: BREAKING $HASHTAG$ crashed could be Germanwings flight $HASHTAG$ from Barcelona to Dusseldorf. URL

After removing stop words: BREAKING $HASHTAG$ crashed could Germanwings flight $HASHTAG$ Barcelona Dusseldorf. URL

After removing special symbols: BREAKING HASHTAG crashed could Germanwings flight HASHTAG Barcelona Dusseldorf. URL

After lowercasing: breaking hashtag crashed could germanwings flight hashtag barcelonadusseldorfurl

After stemming: ['break', 'hashtag', 'crash', 'could', 'germanw', 'flight', 'hashtag', 'barcelona', 'dusseldorf', 'url']

After lemmatization: ['break', 'hashtag', 'crash', 'could', 'germanw', 'flight', 'hashtag', 'barcelona', 'dusseldorf', 'url']

Fig 3.2. Pre-processing sample tweet

3.2.3. Feature Engineering

A generic text classification process involves pre-processing, feature extraction and selection followed by training of model and evaluation of the actual and predicted results [72]. The data extracted from social media is unstructured text. To extract relevant features, it must first be pre-processed. After the data is pre-processed, a number of feature extraction techniques can be applied like BOW, TF-IDF and n-grams. It is followed by the feature selection task. Feature selection is a computationally intensive task wherein a subset of the most relevant features is selected from the entire feature set. In order to reduce the processing time and increase the efficiency of the training and evaluation process, only the most relevant features are selected. Thus, feature engineering involves the tasks of feature extraction, feature transformation and feature selection as shown in fig.3.3.The filters and wrappers used in this research are shown in boldface in the figure.

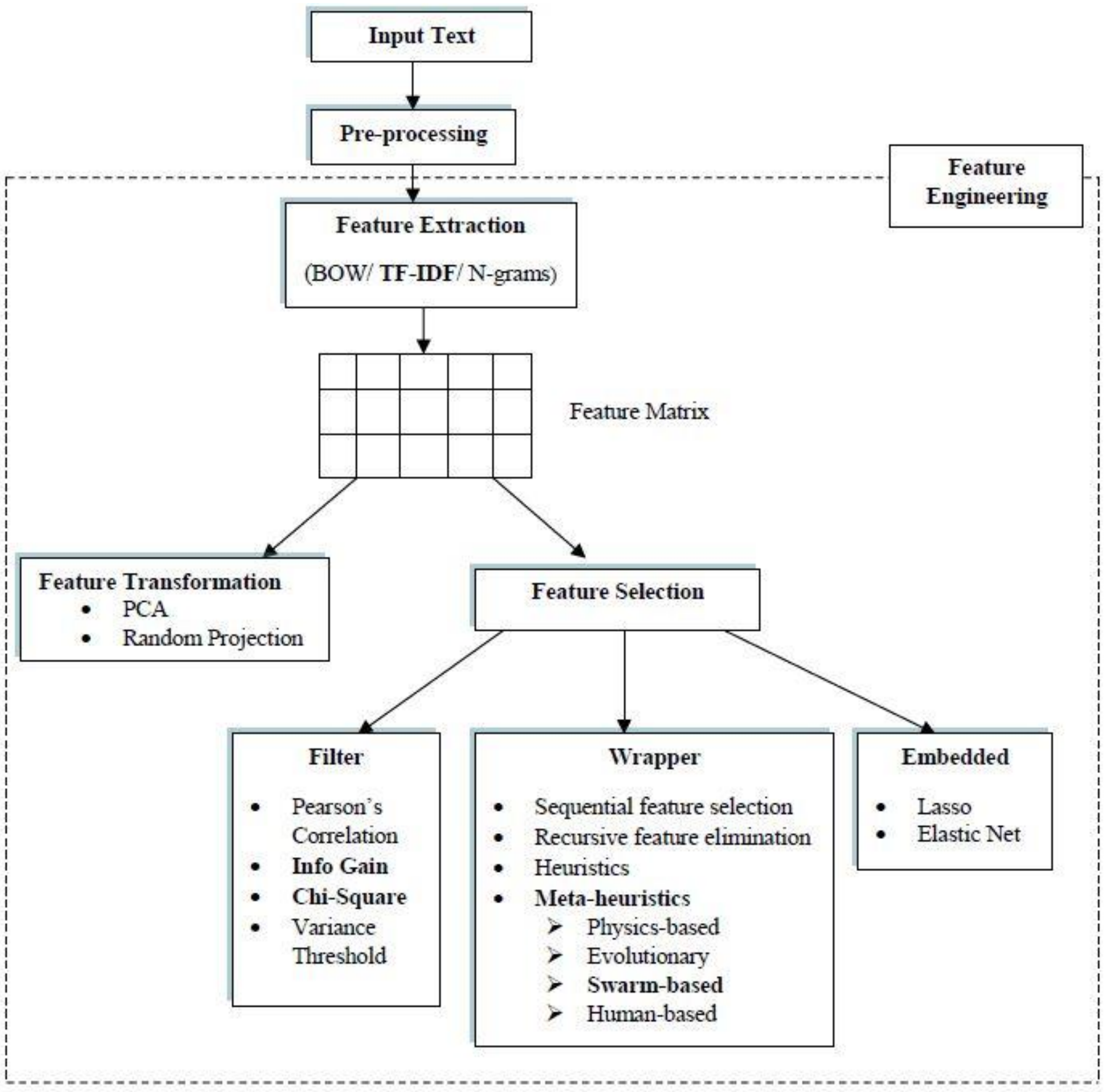


Fig 3.3. Feature Engineering

The following sub-sections elaborate the details of the feature extraction and feature selection approaches used in this work:

3.2.3.1. Feature Extraction

Feature extraction is a crucial step in any text classification task. It is a part of text pre-processing and determines the dimensionality of our dataset. Bag of words, n-grams, and TF-IDF are some of the most widely used feature extraction methods. In this work, we use TF-IDF.

- *Term frequency-inverse document frequency (TF-IDF)*

TF-IDF is a weighting scheme, widely used in text mining tasks [15] [72]. It is a combination of two scores: term frequency and inverse document frequency. The former measures the frequency of occurrence of a particular term in a single document, normalized by dividing with the total number of terms present in that document i.e. the document length to enable comparisons between documents of different lengths. It is denoted by:

$$tf_{t,d} = f_{t,d} / \sum_{t \in d} t \qquad (3.1)$$

where $tf_{t,d}$ denotes the term frequency of term t in document d,
$f_{t,d}$ denotes the frequency of term t in document d and
$\sum_{t \in d} t$ denotes the total number of terms present in document d

Document frequency, on the other hand, measures the number of documents containing a particular term t. Term-frequency gives equal importance to each term in a document, which can be misleading as certain terms like articles and prepositions occur very frequently in documents but are hardly of any relevance. This shortcoming is overcome by using the inverse document frequency which scales down the weights of frequent terms and gives more importance to not so frequent terms. It is calculated as follows:

$$\text{idf}_t = \log \frac{N}{\text{dft}} \qquad (3.2)$$

where idf_t refers to the inverse document frequency of term t
N is the total number of documents in the corpus
df_t is the document frequency of a term t across the corpus
The TF-IDF value is then computed for each term as follows:

$$\text{TF-IDF} = \text{tf}_{t,d} \text{ x } \text{idf}_t \qquad (3.3)$$

Thus, TF-IDF is the measure of how important a word is to a document. Moreover, it checks how relevant the keyword is throughout the corpus [15]. The TF-IDF score is lowest for terms that occur frequently in almost all documents, highest for terms that occur frequently in a small number of documents and in between for terms that occur a few times in some documents or occur in many documents.

3.2.3.2. Feature Selection

Feature selection is an imperative step in any machine learning task, wherein a subset of most relevant features is selected from the entire feature set [73]. It selects an optimal subset of features with the aim of maximizing or minimizing an objective function. Selecting the best subset with the most relevant and minimum number of features is NP-hard and computationally extensive. In this work, we explore five feature selection methods broadly classified into two categories – filter and wrapper as explained below:

- **Filter Methods**

Filter methods are used for selecting a subset of features from the given feature set. This feature subset is selected by assigning a numerical value to each feature in the original feature set. Based on the value, the features can be ranked in order of relevancy and the most relevant ones can be selected for training the classifier. That is, the filter methods pick up the intrinsic properties of the features (i.e., the "relevance" of the features) measured via univariate statistics. A number of filter methods are available out of which two have been used in this work, namely, information gain [74] and chi-square [75]. The two have been briefly described below:

- *Information Gain (IG):* It is a method for calculating the relevancy of a particular feature for the determination of the class label. It measures the information gained in predicting a class value when a particular feature is present or absent. It is based on the concept of entropy and can be defined as a measure of the reduction in entropy of the class variable after the value for the feature is observed. It can be calculated as:

$$\text{IG(t)} = -\sum_{i=1}^{m} p(ci)\log p(ci) + p(t)\sum_{i=1}^{m} p(ci|t)\log p(ci|t) + p(t')\sum_{i=1}^{m} p(ci|t')\log p(ci|t') \qquad (3.4)$$

where c_i indicates the ith class
$p(c_i)$ indicates the probability of the ith class
p(t) and p(t') are respectively the probabilities of presence and absence of the feature t.
$p(c_i|t)$ and $p(c_i|t')$ are the respectively the conditional probabilities given the presence and absence of the feature t.

- *Chi-square (CS):* The Chi-square test is a statistical test that tests the independence between two events namely the presence of a feature and the value of the class variable in the case of feature selection. The Chi square value is calculated between each feature and the target class variable and the desired numbers of features with the top scores are selected. The basic idea being that if a feature is independent of the target variable, it is relatively uninformative and does not contribute much towards class prediction. It is calculated as follows :

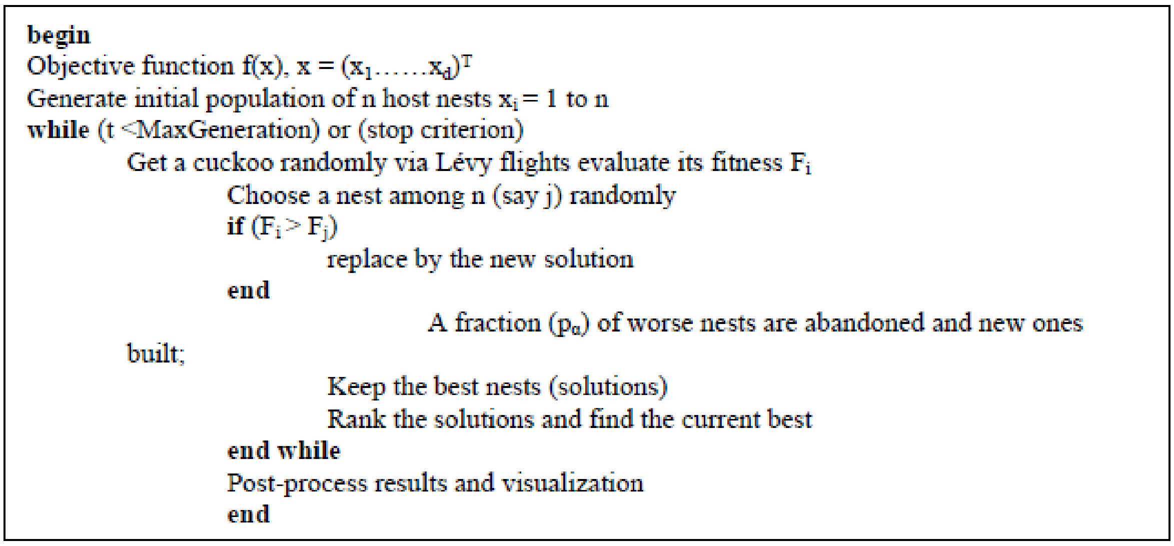
$$\chi^2 = \sum_{i=1}^{n} \frac{(\mathrm{Oi}-\mathrm{Ei})2}{\mathrm{Ei}} \qquad (3.5)$$

where O_i = Observed value
E_i= Expected value

- **Wrapper methods**

In contrast to the filter methods, wrapper methods measure the "usefulness" of features based on the classifier performance. Given the large number of attributes, it is imperative to select the relevant few to shorten training time, enhance generalizability of the model by avoiding overfitting, get simplified models and to avoid the curse of dimensionality. Swarm algorithms are a class of population based meta-heuristics which arrive at an optimum solution using a set of collective, decentralized, distributed and self-organizing agents. The most prominent among the swarm-based algorithms are those inspired by the behaviour of species in nature like birds, ants, insects, etc. In this paper, we use three swarm intelligence based wrapper algorithms as the search methods for finding an optimal feature subset namely cuckoo search, bat, and ant colony optimization.

- *Cuckoo Search:* Proposed by Yang et al. [76] in 2009, the algorithm is inspired by the artificial brood parasitic behaviour of some cuckoo birds wherein the cuckoo birds leave their eggs in the nest of another bird having similar looking eggs to be raised by them. They carefully choose a nest and the cuckoo's eggs if they hatch first; they often kick out the eggs of the host bird to avoid competition for food. On the other hand, if the host bird identifies the egg not to be hers, it either throws the egg away or abandons the nest. The cuckoo search algorithm is based on this brood parasitism and the Lévy flight behaviour of some birds found in nature. The pseudo-code for cuckoo search is given in fig.3.4.



```
begin
Objective function f(x), $x = (x_1......x_d)^T$
Generate initial population of n host nests $x_i$ = 1 to n
while (t <MaxGeneration) or (stop criterion)
          Get a cuckoo randomly via Lévy flights evaluate its fitness $F_i$
                    Choose a nest among n (say j) randomly
                    if ($F_i > F_j$)
                              replace by the new solution
                    end
                                        A fraction ($p_a$) of worse nests are abandoned and new ones
          built;
                              Keep the best nests (solutions)
                              Rank the solutions and find the current best
                    end while
                    Post-process results and visualization
                    end
```

Fig 3.4 Cuckoo Search Pseudo Code

- *Bat Algorithm*: Yang [77] proposed the bat algorithm in 2010 is based on the echolocation behaviour of bats. Echo location, is used by microbats for hunting and avoiding obstacles who emit an ultrasonic sound pulse ranging from 25kHz to 150 kHz and lasting between 5 to 20 ms which bounces back from surrounding objects. The distance between the bat and the object can be estimated by the time taken by the echo to reach the bat. The pseudo-code for bat search is given in fig.3.5.

```
begin
Objective function f(x), x = (x1......xd)^T
Generate initial population of n bats xi = 1 to n
Define pulse frequency fi at xi
Initialize pulse rates ri and loudness Ai
while (t <MaxIterations)
        Generate new solutions by adjusting frequency and updating velocities and locations
                if (rand >ri)
                        Select a solution among the best.
                        Generate a local solution around the selected best solution
                end
                        Generate a new solution by flying randomly
                if(rand < Ai and f (xi) < f(x*))
                        Accept the new solution
                        Increase ri and reduce Ai
                end if
                        Rank the best and find the current bestx*
                end while
                Postprocess results and visualization
                end
```

Fig 3.5. Bat Algorithm Pseudo Code

- *Ant Colony Optimization:* Given by Morco [78] in 1992, it is inspired by the communication process used by ants. Ants, when searching for food start off randomly in a direction. On finding food, the ant returns to its colony leaving pheromone (a chemical) trails on the way back. The pheromone is made stronger if other ants follow the path and find food as well. On the other hand, the trail becomes fainter as it evaporates over time if the path is not travelled by other ants. The pseudo-code for ACO is given in the following fig.3.6.

```
begin
        Initialize pheromone and other parameters
        Generate a population of n ants
        for(ant $_i$)
                Calculate fitness value
                Determine best position
        Determine best global ant (solution)
        Update pheromone trail
        Check stopping criterion
end
```

Fig 3.6. Ant Colony Optimization Pseudo Code

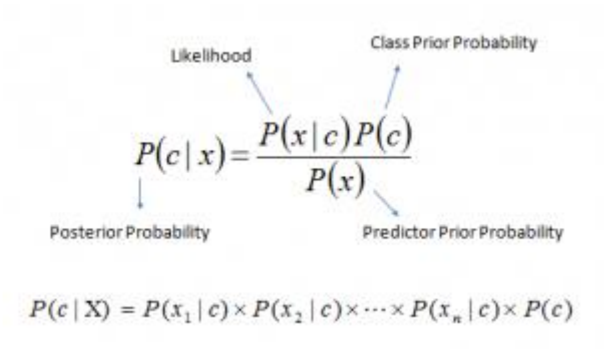
We examine various permutations and combinations of these filters and wrappers to build an optimal rumour detection model and empirically analyze the performance of various feature selection models based on source tweet’s textual features.

3.2.4. Classification

Classification in machine learning (ML) is the task of segregating instances of input into various classes based on previous training input. It is a supervised task, wherein a number of labelled training instances are given to the ML classifier as input. The classifier learns from the training sample. Then the performance is tested on a sample of test instances. The data with reduced feature subset is run through three ML classifiers namely Naïve Bayes (NB), Random Forest (RF) and J48 decision tree (DT) [75], [79]. 10-fold cross validation method is used for evaluating the performance of the various algorithms.

The ML algorithms used are briefly explained below:

- *Naïve Bayes:* It is a classification technique based on Bayes' Theorem with the "naive" assumption of independence among predictors, i.e. the various features are considered to be independent and unrelated to each other. Naïve Bayes is fast to build and works well with large datasets. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig 3.7. Calculating Probability using Naïve Bayes

- *Random Forest:* Random Forest, grows a number of trees. Each tree provides the classification for every single instance and other trees vote for that class. The class with the highest votes is assigned to that instance. A sample of the N input instances is taken randomly with replacement. For M input variables, m<M is selected so that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node.

- *Decision Tree:* It is a supervised learning algorithm that can be used for both discrete and non-discrete variables. The population is split into two or more or sub-populations based on most significant splitter / differentiator in input variables. The data set is broken into subsets having smaller size with decision and leaf nodes.

CHAPTER 4

PROPOSED MODEL FOR RUMOUR ORIGIN

This chapter describes in detail the proposed model for the subtask of rumour origin based on certain meta-features derived from user account and tweet based features.

4.1 Introduction

Identifying potential rumour mongers is imperative to prevent further rumour spread. Moreover, this step can be very helpful in calculating the veracity of a rumour which is the final step in rumour analytics pipeline.

4.2 The proposed PROD: Potential Rumour Origin Detection Model

Fig.4.1 depicts the system architecture of the work undertaken. It consists of four primary modules: Data acquisition, feature extraction, classification and evaluation. These module are expounded in detail in the subsequent sections.

4.2.1 Data Acquistion

The dataset used in this work is the PHEME dataset of rumours. It consists of labelled tweets for 5 breaking news events, namely, #charliehebdo (458 rumours and 1621 non-rumours), #ferguson (284 rumours and 859 non-rumours), #germanwingscrash (238 rumours and 231 non-rumours), #ottawashooting (470 rumours and 420 non-rumours) and #syd-neysiege (522 rumours and 699 non-rumours). As the primary aim of this work is to identify potential rumour source, we only considered rumourous tweets from the dataset for all the five events.

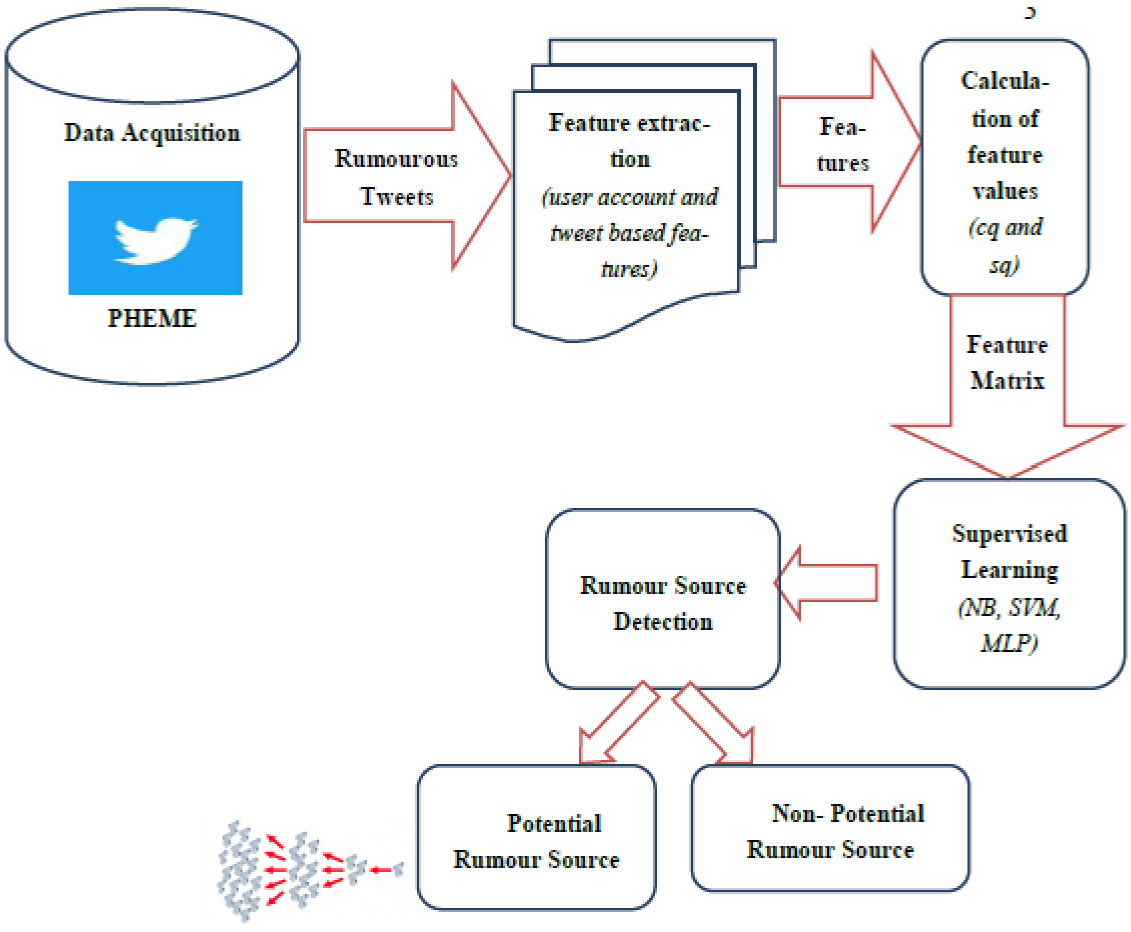


Fig 4.1. Systematic flow of the model

4.2.2 Feature Extraction

In this work we use meta-features derived from user account and tweet based features to detect if a particular user is a rumour source or not. The features extracted are: Date of tweet posting; Date of account creation; Retweet status of post; Retweet count of post; Followers count of user; Status count of user; Favourite count of user; User account verification status; url presence; media presence. Thus, the resulting feature tuple is given as <naiv, orig, resp, activ, sr, av, tv, media>, where, each term is assigned a weight, indicating how negatively or positively each value affects the credibility quotient of the user. Each of these features used to build the PROD model are described in the following table 4.1.

Table 4.1. Features to build the PROD model

Feature	**Description**
<naiv>	It is indicative of user naivety, and depends on the account age of the user. The younger the user account age, more naïve he is and hence more prone to spread rumours. Naivety has been assigned a score of -2.
<orig>	It indicates whether a particular post is actually composed by the user or not. If the post is retweeted, orig is assigned a value of 0 as this user cannot be a rumour source. It has been assigned a score of 1.
<resp>	It is short for responsibility, and is set to 1 if the user has posted a url in his tweet. Posting url is considered to be indicative of genuine news and hence the user cannot be a potential rumour source. It has been assigned a score of 4.
<activ>	It indicates how active a user is and is measured as the ratio of number of status updates to the no. of days since account

	creation. A study [80] shows that active users are more likely to spread rumours. Hence, this parameter has been assigned a score of -2.
<sr>	sr or social reputation is calculated as the ratio of #favourite_count of a user to the #followers of that user and represents how popular a user is.
<av>	It signifies the account verification status of a person. A verified account, usually of public figures, is less likely to be a rumour source. This factor is thus assigned a positive weight of 5. <av> in combination with <sr> has been assigned a negative score of 2 considering that an account which is not verified but has high popularity can be rumour source.
<tv>	It is short for tweet virality. A viral tweet from an unverified and popular account has very high chances of being a rumour source. Hence this feature in combination with <av> and <sr> has been assigned a score of (-2)
<media>	Many rumours are spread, when people use images from a different context in situations where they would seemingly fit, hence misleading people. This factor, thus has also been assigned a negative score of 2.

All these feature values are summed up to calculate the *credibility quotient (cq)* of a user which would range from [-10, 10]. The higher the credibility, less likely is he to be a rumour source. Hence, *the source quotient (sq)* is calculated to be the inverse of the credibility quotient.

4.2.3 The Proposed Algorithm

The pseudo-code for the model is given in fig 4.2.

```
Input: Rumourous tweets
For each post,
Extract tweet_created_at, account_created_at, retweeted, retweet_count, followers_count,
friends_count, status_count, verified, url, media
acc_age = tweet_created_at – account_created_at
if acc_age< 180 then naiv =1
else naiv = 0
if retweeted = False then orig =1
else orig = 0
if url = 1 then resp = 1
else resp = 0
if (statuses_count/followers_count> 4) then activ =1
else activ = 0
if friends_count/followers_count< 0.2 then sr =1
else sr = 0
if verified = true then av =1
else av =0
if retweets > 500 then tv =1
else tv = 0
if media = 1 then media =1
else media = 0
cq =[ orig * 1 + resp * 4 + activ * (-2) + av * 5 + (not(av) and sr) * (-2) + (tv and  not(av) and sr) * (-
2) + media * (-2) + naiv * (-2)]
source_quotient = 1/cq
if (source_quotient<= 0) or (source_quotient> 0.2 and source_quotient< 1), then
prs =1
else prs = 0
    end for
```

Fig 4.2. Pseudo Code of the proposed PROD model

The final value of the source quotient is used to calculate the prs (potential rumour source) binary value, which is also the class label for our dataset. A negative value of source quotient indicates high chances of being a rumour source. Credibility points anywhere between the values 5 to 10 are considered to indicate that concerned user cannot be a potential rumour source. While a credibility score from 1 to 4 is considered to be a rumour origin suspect. Accordingly, the final prs value is labelled as 1 or 0.

4.2.4 Classification

The model is trained using supervised learning wherein a number of labelled train-ing instances are given to the ML classifier Naïve Bayes (NB), Support Vector Machine (SVM) and Multilayer Perceptron (MLP) as input. 10-fold cross validation method is used for evaluating the performance of the various algorithms. ZeroR has been used as the baseline classifier which simply predicts the majority class for all instances. These algorithms are briefly explained below:

- *ZeroR:* The simplest classification method, ZeroR simply relies on the target, ignoring all predictors. It simply predicts the majority class as output. It is a useful baseline classifier. The baseline classification can be used as a benchmark for other algorithms.

- *Naïve Bayes:* It is a classification technique based on Bayes' Theorem with the "naive" assumption of independence among predictors, i.e. the various features are considered to be independent and unrelated to each other. Naïve Bayes is fast to build and works well with large datasets. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c) as already explained in chapter 3.

- *Support Vector Machines:* Support Vector Machine" (SVM) is a supervised machine learning algorithm that finds a separating decision surface, called a hyperplane, to segregate instances of the classes in the dataset. It aims to maximize the distance between the hyperplane and the closest data points of the two classes. This distance is called the margin. Greater the margin lower the classification error.

- *Multilayer Perceptron:* MLP is a feedforward deep artificial neural network with a minimum of three layers – a layer for input, one or more hidden layers and a layer for output. It is the hidden layer neurons that provide the computational capability to the network. It uses backpropagation to adjust weights based on the error, calculated as the difference between the target and the actual output values.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter presents the comprehensive results based on the models presented in the previous two chapters followed by a discussion of the same.

5.1 Results for the Hybrid Filter Wrapper Model

This section discusses the results of various combinations of the filter and wrapper methods with TF-IDF. The results are broadly divided into four subparts on the basis of: (i) performance; (ii) population size of the various swarm algorithms; (iii) magnitude of feature reduction (iv) time taken to build the model.

5.1.1 Basis of Performance

Accuracy, precision, recall and F-measure [15] are the common performance measures used to evaluate the performance of ML classifiers. The results are presented in the form of graphs for the #germanwingscrash event. The results for the remaining events are presented in the form of tables. The average accuracy achieved by individual feature selection methods (2 filters and 3 wrappers) with TF-IDF feature extraction is shown in fig.5.1. Swarm-based wrapper method, cuckoo search shows the highest average accuracy of 60.2% as compared to all the others.

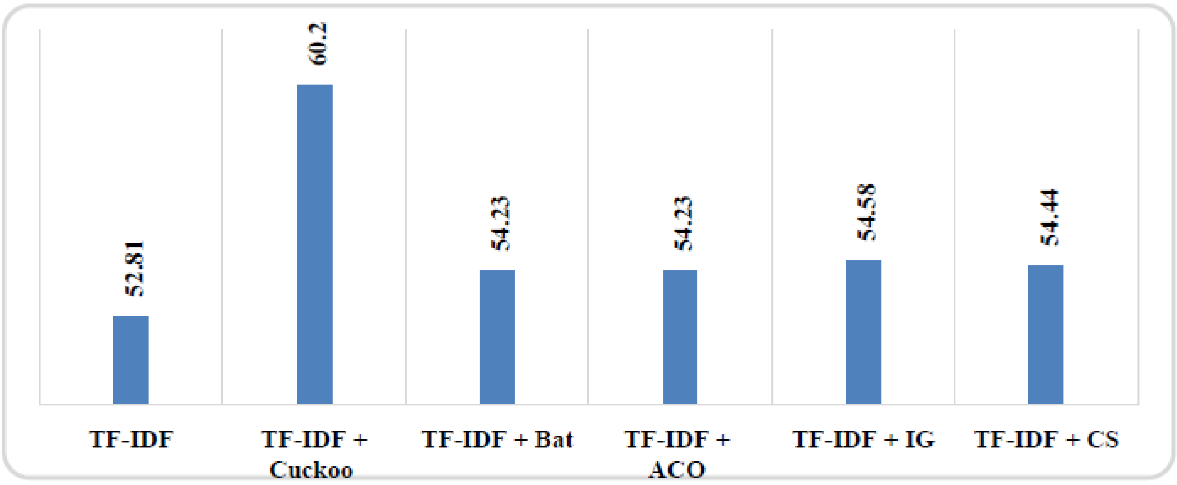


Fig 5.1. Average Accuracy with individual feature selection methods with TF-IDF

For the hybrid of filter and wrappers, the highest accuracy of 61.19% was observed for the combination of chi-square filter with both cuckoo search and ACO wrappers (Fig.5.2).

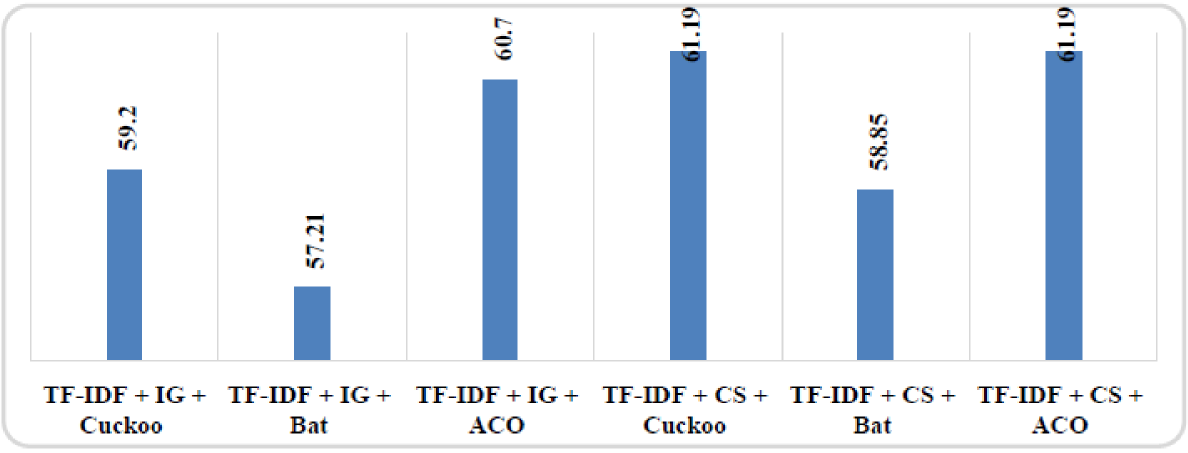


Fig 5.2. Average Accuracy using hybrid filter-wrapper with TF-IDF

The results of using hybrid filter-wrapper for all five events of PHEME dataset are given in tables 5.1 and 5.2.

Table 5.1: Performance with TF-IDF + IG filter+ All Wrappers

GERMAN WINGS CRASH	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}
NB	62.90	0.65	0.63	0.62	52.24	0.754	0.52	0.37	66.52	0.67	0.67	0.67
RF	57.36	0.73	0.58	0.50	62.0469	0.67	0.62	0.60	58.21	0.64	0.58	0.54
DT	57.36	0.60	0.57	0.55	57.3561	0.60	0.57	0.55	57.36	0.60	0.57	0.55
CHARLIE HEBDO	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}
NB	78.11	0.75	0.78	0.69	78.36	0.77	0.78	0.70	78.11	0.79	0.78	0.69
RF	78.50	0.79	0.79	0.70	78.50	0.79	0.79	0.70	78.50	0.79	0.79	0.70
DT	77.97	0.78	0.78	0.72	77.97	0.78	0.78	0.70	77.97	0.79	0.78	0.72
FERGUSON UNREST	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			

Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	75.24	0.71	0.75	0.65	75.07	0.67	0.75	0.65	75.77	0.76	0.76	0.66
RF	75.85	0.79	0.76	0.66	75.85	0.79	0.76	0.66	75.85	0.79	0.76	0.66
DT	75.77	0.82	0.76	0.66	75.59	0.82	0.76	0.66	75.59	0.82	0.76	0.66
<u>OTTAWA SHOOTING</u>	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	68.31	0.70	0.69	0.68	62.02	0.70	0.62	0.59	61.91	0.67	0.62	0.60
RF	60.11	0.76	0.60	0.54	67.19	0.68	0.67	0.67	67.42	0.68	0.67	0.67
DT	54.05	0.75	0.54	0.39	53.82	0.75	0.54	0.39	54.04	0.75	0.54	0.39
<u>SYDNEY SEIGE</u>	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	58.07	0.58	0.58	0.47	58.56	0.64	0.59	0.46	58.97	0.64	0.59	0.47

RF	58.72	0.72	0.59	0.45	58.72	0.72	0.59	0.45	58.72	0.72	0.59	0.45
DT	57.58	0.69	0.58	0.43	57.58	0.69	0.58	0.43	57.66	0.70	0.58	0.43

Table 5.2: Performance with TF-IDF + Chi-Square filter+ All Wrappers

GERMAN WINGS CRASH	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	65.46	0.67	0.65	0.64	59.70	0.61	0.60	0.59	66.10	0.69	0.66	0.65
RF	60.77	0.61	0.61	0.61	59.49	0.71	0.60	0.54	60.13	0.66	0.60	0.57
DT	57.36	0.59	0.57	0.55	57.36	0.59	0.574	0.55	57.36	0.59	0.57	0.55
CHARLIE HEBDO	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	78.16	0.79	0.78	0.69	78.02	0.74	0.78	0.69	78.21	0.83	0.78	0.69

RF	78.50	0.79	0.79	0.70	78.50	0.79	0.79	0.70	78.50	0.79	0.79	0.70
DT	78.11	0.83	0.78	0.69	77.97	0.77	0.78	0.72	77.97	0.80	0.78	0.69
FERGUSON UNREST	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}
NB	75.42	0.77	0.75	0.65	75.68	0.73	0.76	0.67	76.03	0.78	0.760	0.67
RF	75.85	0.79	0.76	0.66	75.85	0.79	0.76	0.66	75.85	0.79	0.76	0.
DT	75.59	0.82	0.76	0.66	75.50	0.82	0.76	0.65	75.50	0.82	0.76	0.65
OTTAWA SHOOTING	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}	A_c	P_r	R_e	F_{meas}
NB	59.33	0.68	0.60	0.55	59.66	0.65	0.60	0.57	49.55	0.71	0.50	0.36
RF	52.70	0.744	0.53	0.42	61.80	0.71	0.62	0.59	51.35	0.76	0.51	0.39

DT	54.05	0.75	0.54	0.39	54.05	0.75	0.54	0.39	54.05	0.75	0.54	0.39
<u>SYDNEY SEIGE</u>	**CUCKOO SEARCH**				**BAT ALGORITHM**				**ANT COLONY OPTIMIZATION**			
Classifier	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**	**A_c**	**P_r**	**R_e**	**F_{meas}**
NB	59.13	0.67	0.59	0.47	59.21	0.71	0.59	0.47	59.46	0.63	0.60	59.46
RF	58.72	0.72	0.59	0.45	58.72	0.72	0.59	0.45	58.72	0.72	0.59	58.72
DT	57.82	0.71	0.58	0.43	57.74	0.67	0.58	0.43	57.82	0.71	0.58	57.82

5.1.2 Population size

Population in swarm-based algorithms refers to the number of agents looking for solutions. We compared the performance (accuracy) for three population sizes – 10, 15 and 20 for all the three swarm algorithms. ACO gives the best results at population size 15 when used with the chi-square filter, whereas cuckoo search gives best results at population size 20 when used with the same chi-square filter, as shown in fig. 5.3 and fig 5.4 respectively.

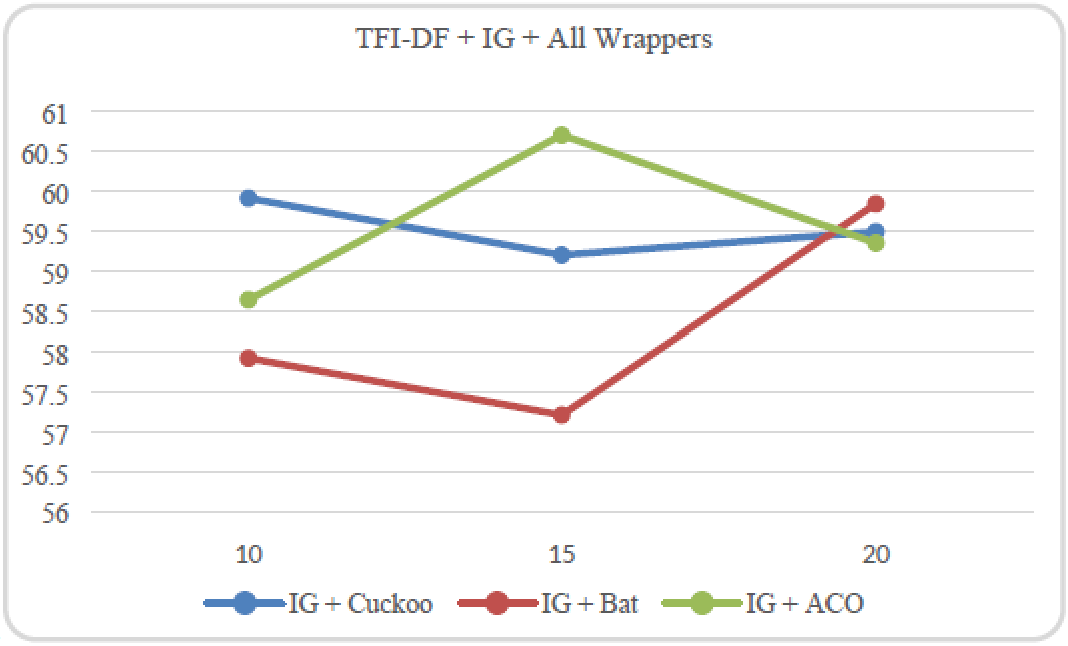


Fig 5.3. Population size comparison of all wrappers with information gain

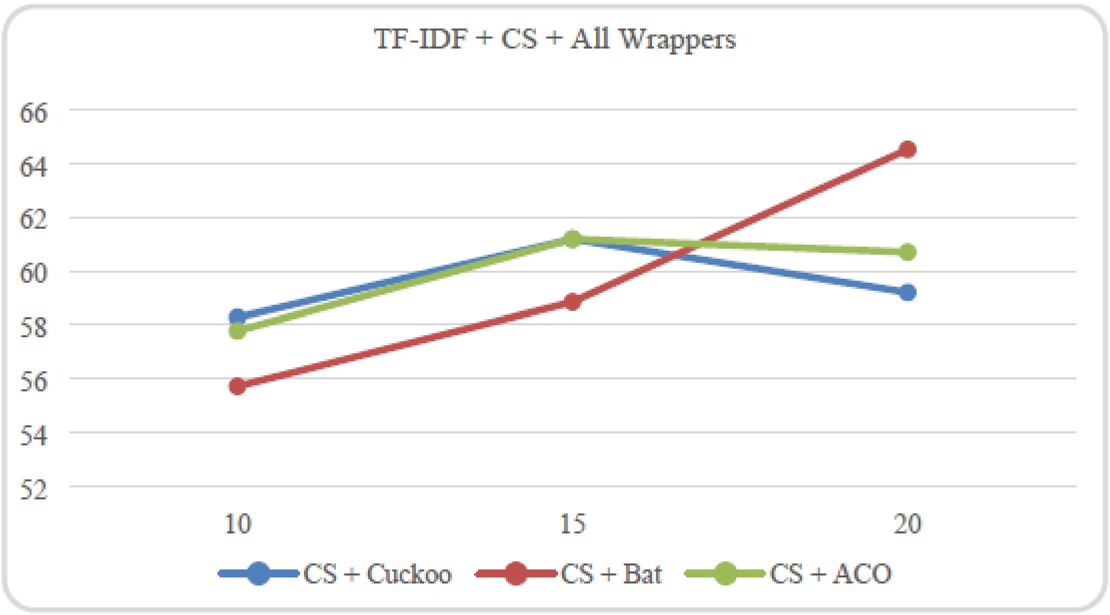


Fig 5.4. Population size comparison of all wrappers with Chi-Square

5.1.3. Feature reduction

In this section, we compare the performance of the optimization algorithms based on the number of features they return in the optimal subset. To maintain consistency, the algorithms are all compared for population size 15. These results are depicted in figures 5.5 and 5.6. Bat algorithm when used with information gain and chi-square, gives best feature reduction at 81% and 82% respectively.

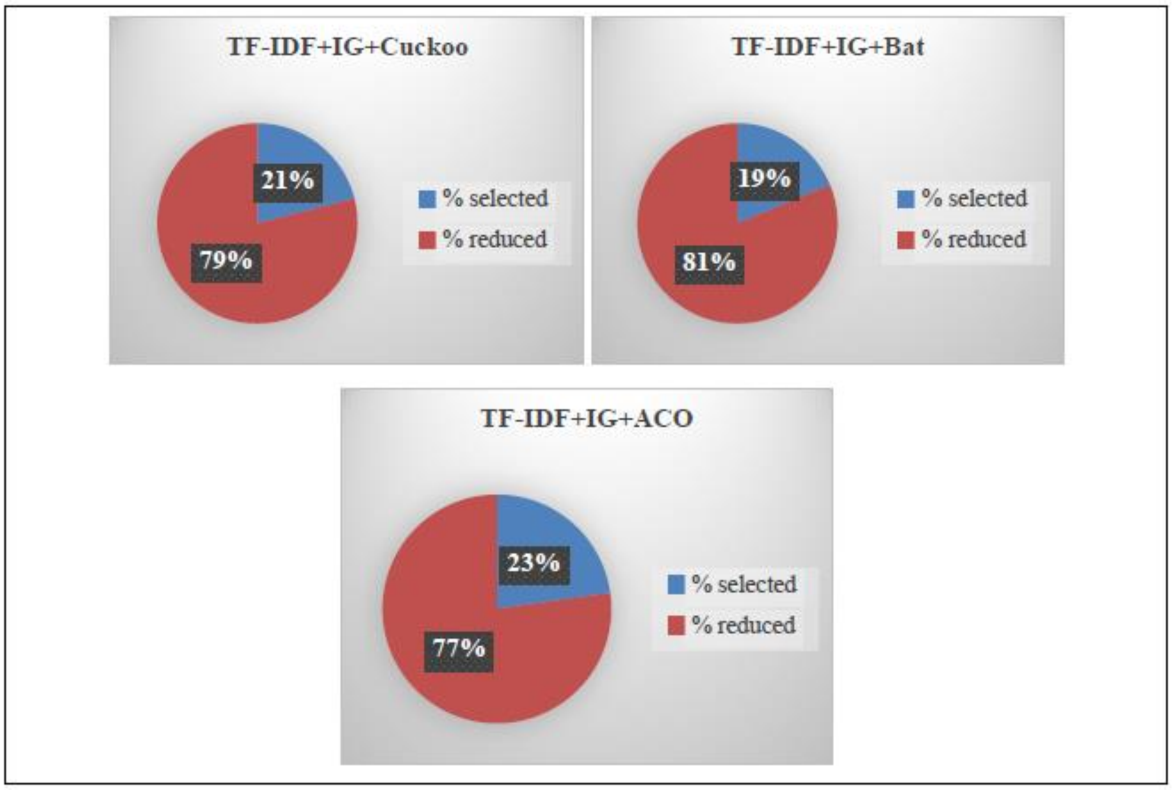


Fig 5.5. Feature Reduction using IG and all wrappers

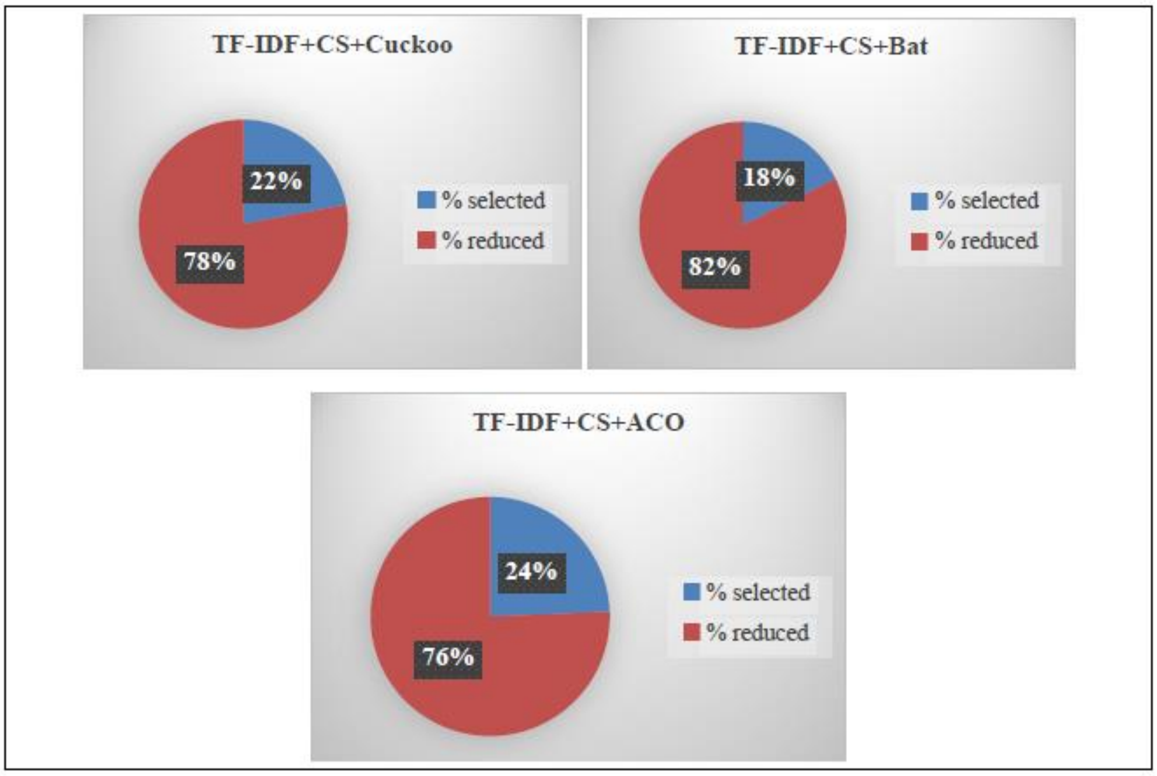


Fig 5.6. Feature Reduction using CS and all wrappers

5.1.4. Time taken to build model

For all combinations, Naïve Bayes takes the least amount of time for model building and random forest takes the maximum time (10 times more than Naïve Bayes).For the hybrid filter-wrapper, chi-square filter used with the bat swarm-based wrapper algorithm takes the least amount of time. The results are depicted in fig. 5.7.

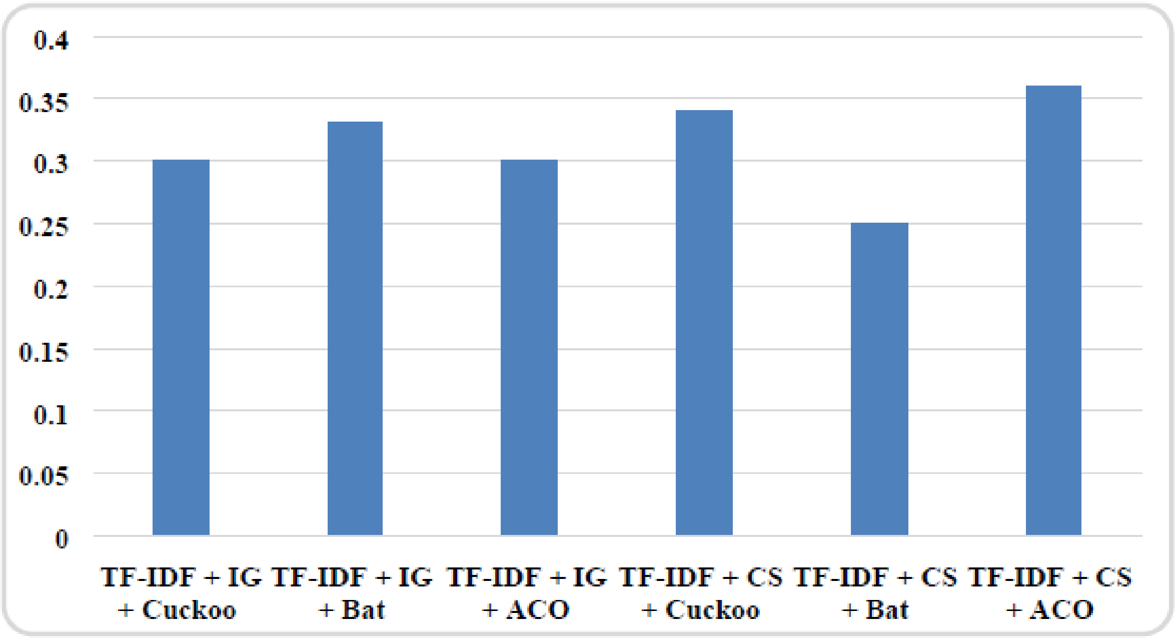


Fig 5.7. Time taken to build model using hybrid filter-wrapper

5.1.5. Discussion

Both cuckoo search and ACO give the highest accuracy when used with CS. Bat Algorithm in combination with CS outperforms cuckoo search and ACO in terms of both feature reduction and time taken to build model. The results of hybrid filter-wrapper are summarized as follows:

Accuracy: CS + Cuckoo = CS + ACO > IG + ACO > IG + Cuckoo > CS + Bat > IG + Bat

Magnitude of feature reduction: CS + Bat > IG + Bat > IG + Cuckoo > CS + Cuckoo > IG + ACO > CS + ACO

Time taken to build model: CS + ACO > CS + Cuckoo > IG + Bat > IG + Cuckoo = IG + ACO > CS + Bat

5.2. Results for the PROD Model

WEKA tool has been used for evaluation purposes. It is an open source software tool with a collection of machine learning algorithms and tools for classification, visualization, etc. The results for all five events in PHEME dataset have been evaluated for 4 performance measures: accuracy, precision, recall and F-measure [15] and are summarized in table 5.3. The highest accuracy for each even is obtained by MLP for all events followed by SVM and NB. The accuracy result for MLP classifier for all events is shown as a graph in fig 5.8.

Table 5.3: Performance Results for PROD

CHARLIE HEBDO				
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
ZeroR	65.2838	0.653	0.653	0.784
NB	93.2314	0.937	0.932	0.933
SVM	97.5983	0.976	0.976	0.976
MLP	97.5	0.987	0.987	0.987
FERGUSON				
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
ZeroR	82.3944	0.824	0.824	0.903
NB	87.6761	0.872	0.877	0.857
SVM	95.0704	0.951	0.951	0.951

MLP	97.9	0.97	0.97	0.97
GERMAN WINGS CRASH				
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
ZeroR	65.9664	0.660	0.660	0.795
NB	88.2353	0.890	0.882	0.884
SVM	92.437	0.924	0.924	0.924
MLP	95.7983	0.958	0.958	0.958
OTTAWA SHOOTING				
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
ZeroR	70.6383	0.706	0.706	0.828
NB	95.5319	0.955	0.955	0.955
SVM	96.383	0.964	0.964	0.964
MLP	97.234	0.972	0.972	0.972
SYDNEY SEIGE				
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
ZeroR	68.3301	0.683	0.683	0.812
NB	93.666	0.937	0.937	0.937
SVM	95.0096	0.952	0.950	0.951
MLP	97.8887	0.979	0.979	0.979

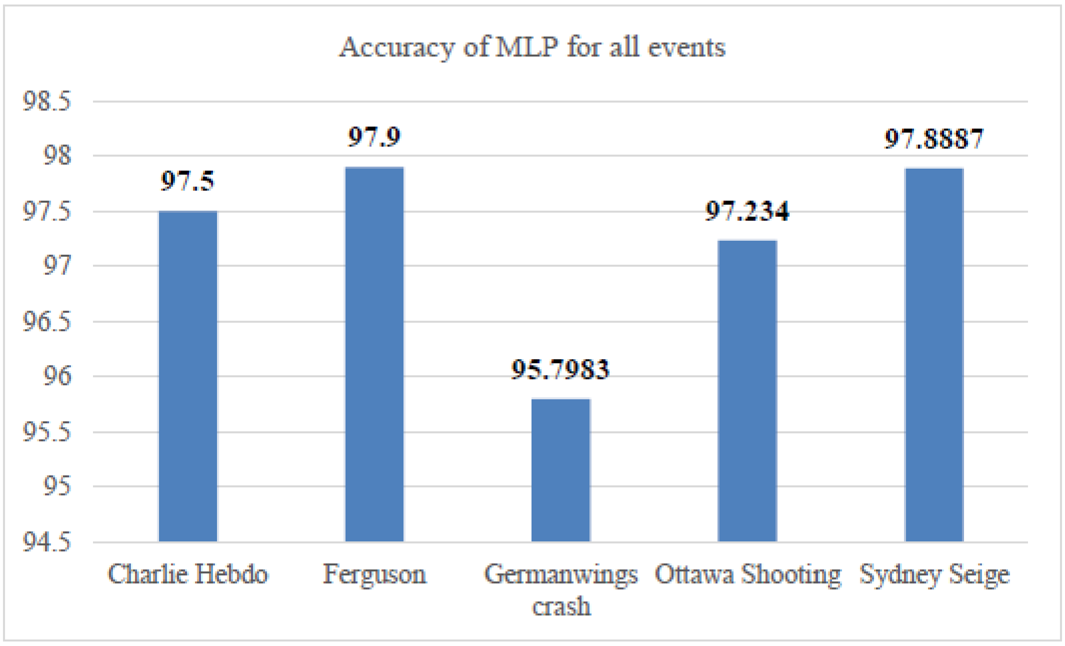


Fig 5.8 Accuracy of MLP for all events

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

This work empirically analysed various combinations of filter and wrapper methods for text-based rumour detection for five events of benchmark PHEME Twitter dataset. TF-IDF was used for feature extraction and feature selection was done using hybrid filter-wrappers. Two filters, chi-square and information gain were combined with three swarm-based wrappers, namely, cuckoo, bat and ACO algorithms. The optimal feature set generated using chi-square with cuckoo and chi-square with ACO yielded the same classifier performance accuracy. Bat algorithm along with the chi-square filter gave the maximum feature reduction of 18% and also took least amount of 0.25 secs to build the model. The results are convincing to affirm that using hybrid filter-wrapper methods facilitates and speeds up the entire pipeline of rumour resolution as the subsequent stages would only look at the most probable candidate rumorous posts. As a future direction, next stages of the rumour resolution pipeline can be explored using the hybrid filter-wrapper model. Also, as this work presents text-based rumour detection, context modelling can be done to improve the detection and debunking of rumourous stories. Moreover, as the social networks are an informal way of communication, a lot of uncertainty in expressiveness exists. The capabilities of neural networks with the abilities of fuzzy logic can be studied to build a model which includes filtering textual comments, user profiles and information propagation patterns to detect rumours. Moreover, location-based [81] and cyber-physical systems [82] [83] [84] at the edge of the Internet can be used as contextual cues for enhanced rumour analytics.

This work also proposed a PROD model for detecting users that can be probable rumour sources. An 8 tuple feature vector has been used to evaluate the credibility of each user. The likelihood of the user being a potential rumour source is the inverse of the obtained credibility quotient. The resulting value is used to label each user as a potential or non-potential rumour source. The multi-layer perceptron classifier out performs the Naïve Bayes and support vector machine classifiers in terms of performance accuracy, for all five events of PHEME, thus demonstrating the superior use supervised learning for potential rumour source detection model. A limitation of the proposed model is that only few features have been used to detect a potential rumour origin. The model currently does not also detect rumour source for end-to-end encrypted social media such as WhatsApp. This work can be extended to include more meta- features and using fuzzy values instead of binary values. Also, an optimization on the feature set can be carried out to determine the most relevant features for rumour origin prediction.

APPENDIX A

List of Publications:

- Kumar, A. and Sharma, H., “PROD: A Potential Rumour Origin Detection Model using Supervised Machine Learning,” Proceedings of Conference on Intelligent Computing and Smart Computation, April. 19-21, PAPERID-360. Springer (2019) (Accepted)
- Kumar, A. and Sharma H., “Hybrid Filter-Wrapper Feature Selection for Rumour Detection on PHEME dataset,” Internet of Things: Engineering Cyber Physical Human Systems. Elsevier (Communicated)

BIBLIOGRAPHY

[1]. Kumar, A., & Sachdeva, N. (2019). Cyberbullying detection on social multimedia using soft computing techniques: a meta-analysis. Multimedia Tools and Applications, 1-38.

[2]. Kumar, A., Nayak, S., & Chandra, N. (2019). Empirical Analysis of Supervised Machine Learning Techniques for Cyberbullying Detection. In International Conference on Innovative Computing and Communications (pp. 223-230). Springer, Singapore.

[3]. Chen, W., & Lee, K. H. (2013). Sharing, liking, commenting, and distressed? The pathway between Facebook interaction and psychological distress. Cyberpsychology, behavior, and social networking, 16(10), 728-734.

[4]. Baker, J. R., & Moore, S. M. (2008). Distress, coping, and blogging: Comparing new Myspace users by their intention to blog. CyberPsychology&Behavior, 11(1), 81-85.

[5]. Garimella, K., De Francisci Morales, G., Gionis, A., &Mathioudakis, M. (2018, April). Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In Proceedings of the 2018 World Wide Web Conference on World Wide Web (pp. 913-922). International World Wide Web Conferences Steering Committee.

[6]. Kumar, A., Sangwan,S R. (2018). Information Virality Prediction using Emotion Quotient of Tweets. International Journal of Computer Sciences and Engineering 6(6), 642-651.

[7]. Kumar, A., Sangwan, S. R., & Nayyar, A. (2019). Rumour veracity detection on twitter using particle swarm optimized shallow classifiers. Multimedia Tools and Applications, 1-19.

[8]. Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146-1151.

[9]. Donovan, P. (2007). How idle is idle talk? One hundred years of rumor research. Diogenes, 54(1), 59-82.

[10]. DiFonzo, N., & Bordia, P. (2007). Rumor, gossip and urban legends. Diogenes, 54(1), 19-35.

[11]. Allport, F. H., & Lepkin, M. (1945). Wartime rumors of waste and special privilege: why some people believe them. The Journal of Abnormal and Social Psychology, 40(1), 3.

[12]. Allport, G. W., & Postman, L. (1947). The psychology of rumor.

[13]. Zubiaga, A., Liakata, M., Procter, R., Hoi, G. W. S., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one, 11(3), e0150989.

[14]. Hsu, H. H., Hsieh, C. W., & Lu, M. D. (2011). Hybrid feature selection by combining filters and wrappers. Expert Systems with Applications, 38(7), 8144-8150.

[15]. Bhatia, M. P. S., & Khalid, A. K. (2008). A Primer on the Web Information Retrieval Paradigm. Journal of Theoretical & Applied Information Technology, 4(7).

[16]. Yang, F., Liu, Y., Yu, X., & Yang, M. (2012, August). Automatic detection of rumor on Sina Weibo. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (p. 13). ACM.
[17]. Zhang, Q., Zhang, S., Dong, J., Xiong, J., & Cheng, X. (2015). Automatic detection of rumor on social network. In Natural Language Processing and Chinese Computing (pp. 113-122). Springer, Cham.
[18]. Jin, Z., Cao, J., Jiang, Y. G., & Zhang, Y. (2014, December). News credibility evaluation on microblog with a hierarchical propagation model. In 2014 IEEE International Conference on Data Mining (pp. 230-239). IEEE
[19]. Wang, S., &Terano, T. (2015, October). Detecting rumor patterns in streaming social media. In 2015 IEEE International Conference on Big Data (Big Data) (pp. 2709-2715). IEEE
[20]. Yang, Y., Niu, K., & He, Z. (2015, July). Exploiting the topology property of social network for rumor detection. In 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE) (pp. 41-46). IEEE
[21]. Kumar, A., Khorwal, R., & Chaudhary, S. (2016). A survey on sentiment analysis using swarm intelligence. Indian J SciTechnol, 9(39), 1-7.
[22]. Kumar, A., & Garg, G. (2019). Sentiment analysis of multimodal twitter data. Multimedia Tools and Applications, 1-17.
[23]. Kumar, A., & Garg, G. (2019). Systematic literature review on context-based sentiment analysis in social multimedia. Multimedia Tools and Applications, 1-32.
[24]. Kumar, A., & Jaiswal, A. (2019). Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on twitter. Multimedia Tools and Applications, 1-25.
[25]. Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model with Convolution Network. IEEE Access.
[26]. Kumar, A., & Ahmad, N. (2012). ComEx miner: Expert mining in virtual communities. In-ternational Journal of Advanced Computer Science and Applications (IJACSA), 3(6).
[27]. Kumar, A., &Sangwan, S. R. (2019). Rumour Detection Using Machine Learning Techniques on Social Media. In International Conference on Innovative Computing and Communications (pp. 213-221). Springer, Singapore.
[28]. Procter, R., Crump, J., Karstedt, S., Voss, A., and Cantijoch, M. (2013a). Reading the riots: What were the police doing on twitter? Policing and society, 23(4):413-436
[29]. Procter, R., Vis, F., and Voss, A. (2013b). Reading the riots on twitter: methodological innovation for the analysis of big data. International journal of social research methodology, 16(3):197-214.
[30]. Takahashi, T. and Igata, N. (2012). Rumor detection on twitter. In Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on, pages 452{457. IEEE.
[31]. Tolosi, L., Tagarev, A., and Georgiev, G. (2016). An analysis of event-agnostic features for rumour classification in twitter. In ICWSM Workshop on Social Media in the Newsroom.
[32]. Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., and Tolmie, P. (2015). Crowdsourcing the annotation of rumourous conversations in social media. In Proceedings

of the 24th International Conference on World Wide Web Companion, pages 347-353.International World Wide Web Conferences Steering Committee.

[33]. Starbird, K., Maddock, J., Orand, M., Achterman, P., and Mason, R. M. (2014). Rumors, false fags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. iConference 2014 Proceedings.

[34]. Seo, E., Mohapatra, P., &Abdelzaher, T.: Identifying Rumours and their sources in social networks. In Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III (Vol. 8389, p. 83891I). International Society for Optics and Photonics (2012)

[35]. Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In Proceedings of EMNLP, pages 1589-1599.

[36]. Hamidian, S., &Diab, M. (2015). Rumor detection and classification for twitter data. In Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS) (pp. 71-77).

[37]. Hamidian, S. and Diab, M. T. (2016). Rumor identification and belief investigation on twitter. In Proceedings of NAACL-HLT, pages 3-8.

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

[39]. Lukasik, M., Cohn, T., and Bontcheva, K. (2015). Classifying tweet level judgements of rumours in social media. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, pages 2590-2595.

[40]. Zeng, L., Starbird, K., and Spiro, E. S. (2016). # unconfrmed: Classifying rumor stance in crisis-related social media messages. In Tenth International AAAI Conference on Web and Social Media.

[41]. Wu, K., Yang, S., and Zhu, K. Q. (2015). False rumors detection on sina weibo by propagation structures. In 2015 IEEE 31st International Conference on Data Engineering, pages 651- 662. IEEE.

[42]. Sun, S., Liu, H., He, J., and Du, X. (2013). Detecting event rumors on sina weibo automatically. In Asia-Pacific Web Conference, pages 120-131. Springer.

[43]. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. F., & Cha, M. (2016, July). Detecting Rumors from Microblogs with Recurrent Neural Networks. In IJCAI (pp. 3818-3824).

[44]. Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K. F. (2015, October). Detect rumors using time series of social context information on microblogging websites. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 1751-1754). ACM.

[45]. Liang, G., He, W., Xu, C., Chen, L., and Zeng, J. (2015). Rumor identification in microblogging systems based on users behavior. IEEE Transactions on Computational Social Systems, 2(3):99-108.

[46]. Jin, Z., Cao, J., Zhang, Y., and Luo, J. (2016, March). News verification by exploiting conflicting social viewpoints in microblogs. In Thirtieth AAAI Conference on Artificial Intelligence.

[47]. Cai, G., Wu, H., and Lv, R. (2014). Rumors detection in chinese via crowd responses. In Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pages 912-917. IEEE.
[48]. Liu, Zhiyuan, et al. "Statistical and semantic analysis of rumors in Chinese social media." Scientia Sinica Informationis 45.12 (2015): 1536.
[49]. Xiao, Renbin, and Tongyang Yu. "A multi-agent simulation approach to rumor spread in virtual commnunity based on social network." Intelligent Automation & Soft Computing 17.7 (2011): 859-869.
[50]. Sahana, V. P., Pias, A. R., Shastri, R., &Mandloi, S. (2015, December). Automatic detection of rumoured tweets and finding its origin. In 2015 International Conference on Computing and Network Communications (CoCoNet) (pp. 607-612). IEEE.
[51]. Zhao, Z., Resnick, P., & Mei, Q. (2015, May). Enquiring minds: Early detection of rumors in social media from enquiry posts. In Proceedings of the 24th International Conference on World Wide Web (pp. 1395-1405). International World Wide Web Conferences Steering Committee.
[52]. Castillo, C., Mendoza, M., & Poblete, B. (2011, March). Information credibility on twitter. In Proceedings of the 20th international conference on World wide web (pp. 675-684). ACM.
[53]. Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013, December). Prominent features of rumor propagation in online social media. In 2013 IEEE 13th International Conference on Data Mining (pp. 1103-1108). IEEE.
[54]. Ma, B., Lin, D., & Cao, D. (2017). Content representation for microblog rumor detection. In Advances in Computational Intelligence Systems (pp. 245-251). Springer, Cham.
[55]. Gupta, M., Zhao, P., & Han, J. (2012, April). Evaluating event credibility on twitter. In Proceedings of the 2012 SIAM International Conference on Data Mining (pp. 153-164). Society for Industrial and Applied Mathematics.
[56]. Giasemidis, G., Singleton, C., Agrafiotis, I., Nurse, J. R., Pilgrim, A., Willis, C., &Greetham, D. V. (2016, November). Determining the veracity of rumours on Twitter. In International Conference on Social Informatics (pp. 185-205). Springer, Cham.
[57]. Kwon, S., Cha, M., & Jung, K. (2017). Rumor detection over varying time windows. PloS one, 12(1), e0168344.
[58]. Jin, Z., Cao, J., Zhang, Y., Zhou, J., & Tian, Q. (2017). Novel visual and statistical image features for microblogs news verification. IEEE transactions on multimedia, 19(3), 598-608.
[59]. Zhang, Z., Zhang, Z., & Li, H. (2015). Predictors of the authenticity of Internet health rumours. Health Information & Libraries Journal, 32(3), 195-205.
[60]. Chen, T., Li, X., Yin, H., & Zhang, J. (2018, June). Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 40-52). Springer, Cham.
[61]. Ruchansky, N., Seo, S., & Liu, Y. (2017, November). Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 797-806). ACM.
[62]. Jin, Z., Cao, J., Guo, H., Zhang, Y., & Luo, J. (2017, October). Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In Proceedings of the 25th ACM international conference on Multimedia (pp. 795-816). ACM.

[63]. Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2017, August). A Convolutional Approach for Misinformation Identification. In IJCAI (pp. 3901-3907).

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

[65]. Shah, D., & Zaman, T.:Rumours in a network: Who's the culprit?. IEEE Transactions on in-formation theory, 57(8), 5163-5181 (2011).

[66]. Dong, W., Zhang, W., & Tan, C. W.: Rooting out the Rumour culprit from suspects. In 2013 IEEE International Symposium on Information Theory (pp. 2671-2675). IEEE (2013).

[67]. Xu, W., & Chen, H. :Scalable Rumour source detection under independent cascade model in online social networks. In 2015 11th International Conference on Mobile Ad-hoc and Sen-sor Networks (MSN) (pp. 236-242). IEEE (2015).

[68]. Król, D., &Wiśniewska, K. : On Rumour source detection and Its experimental verification on twitter. In Asian Conference on Intelligent Information and Database Systems (pp. 110-119). Springer, Cham (2017).

[69]. Wang, Z., Dong, W., Zhang, W., & Tan, C. W. (2014, June). Rumour source detection with multiple observations: Fundamental limits and algorithms. In ACM SIGMETRICS Perfor-mance Evaluation Review (Vol. 42, No. 1, pp. 1-13). ACM.

[70]. Jović, A., Brkić, K., &Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1200-1205). IEEE.

[71]. Jain, D. K., Kumar, A., Sangwan, S. R., Nguyen, G. N., & Tiwari, P. (2019). A Particle Swarm Optimized Learning Model of Fault Classification in Web-Apps. IEEE Access, 7, 18480-18489.

[72]. Kumar, A., Dabas, V., &Hooda, P. (2018). Text classification algorithms for mining unstructured data: a SWOT analysis. International Journal of Information Technology, 1-11.

[73]. Omar, N., Jusoh, F., Ibrahim, R., & Othman, M. S. (2013). Review of feature selection for solving classification problems. Journal of Information System Research and Innovation, 3, 64-70.

[74]. Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

[75]. Bhatia, M. P. S., & Kumar, A. (2008). Information retrieval and machine learning: Supporting technologies for web mining research and practice. Webology, 5(2), 5.

[76]. Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 210-214). IEEE.

[77]. Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.

[78]. Dorigo, M. (1992). Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano.

[79]. Kumar, A., & Jaiswal, A. (2017). Empirical study of Twitter and Tumblr for sentiment analysis using soft computing techniques. In Proceedings of the world congress on engineering and computer science (Vol. 1, pp. 1-5).
[80]. Hindustan times website: https://www.hindustantimes.com/tech/active-twitter-users-most-likely-to-spread-fake-news-study/story-sxrZe611IBYPxv0Pmn8hGO.html
[81]. Meng, S., Qi, L., Li, Q., Lin, W., Xu, X., & Wan, S. (2019). Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems. Future Generation Computer Systems.
[82]. Xu, X., Xue, Y., Qi, L., Yuan, Y., Zhang, X., Umer, T., & Wan, S. (2019). An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Future Generation Computer Systems.
[83]. Wan, S., Zhao, Y., Wang, T., Gu, Z., Abbasi, Q. H., & Choo, K. K. R. (2019). Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. Future Generation Computer Systems, 91, 382-391.
[84]. Din, F. U., Ahmad, A., Ullah, H., Khan, A., Umer, T., & Wan, S. (2019). Efficient sizing and placement of distributed generators in cyber-Physical power systems. Journal of Systems Architecture.