Manchester Metropolitan University

e-space

Manchester Metropolitan University's Research Repository

Please cite the Published Version

Kumar, A, Bhatia, MPS and Sangwan, SR (2022) Rumour detection using deep learning and filter-wrapper feature selection in benchmark Twitter dataset. Multimedia Tools and Applications, 81 (24). pp. 34615-34632. ISSN 1380-7501

DOI: https://doi.org/10.1007/s11042-021-11340-x

Publisher: Springer (part of Springer Nature)

Version: Accepted Version

Downloaded from: https://e-space.mmu.ac.uk/629492/

Usage rights: © In Copyright

Additional Information: This is an Author Accepted Manuscript of an article published in Multimedia Tools and Applications by Springer.

Enquiries:
If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

Rumour Detection using Deep Learning and Filter-Wrapper Feature Selection in Benchmark Twitter Dataset

Akshi Kumar[1], MPS Bhatia[2], Saurabh Raj Sangwan[3]*

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

[2, 3] Department of Computer Science & Engineering, Netaji Subhas University of Technology, New Delhi, India

**saurabh.trf18@nsut.ac.in*

Abstract. Microblogs have become a customary news media source in recent times. But as synthetic text or *'readfakes'* scale up the online disinformation operation, unsubstantiated pieces of information on social media platforms can cause significant havoc by misleading people. It is essential to develop models that can detect rumours and curtail its cascading effect and virality. Undeniably, quick rumour detection during the initial propagation phase is desirable for subsequent veracity and stance assessment. Linguistic features are easily available and act as important attributes during the initial propagation phase. At the same time, the choice of features is crucial for both interpretability and performance of the classifier. Motivated by the need to build a model for automatic rumour detection, this research proffers a hybrid model for rumour classification using deep learning (Convolution neural network) and a filter-wrapper (Information gain –Ant colony) optimized Naive Bayes classifier, trained and tested on the PHEME rumour dataset. The textual features are learnt using the CNN which are combined with the optimized feature vector generated using the filter-wrapper technique, IG-ACO. The resultant optimized vector is then used to train the Naïve Bayes classifier for rumour classification at the output layer of CNN. The proposed classifier shows improved performance to the existing works.

Keywords: classification, deep learning, feature selection, rumour, social media

1. Introduction

As social media is a fertile ground for origin and spread of rumours, it is imperative to detect and deter rumours. A rumour is any chunk of information circulated in the public domain without adequate awareness and confirmation to support its legitimacy [1]. It spreads like wildfire and is believed overtly especially during a crisis situation. Undeniably, the economics of social media favors rumours, hate-speech, pseudo-news, alternative facts or fake news [2-4]. The wave of misinformation and rumour pertaining to the COVID-19 on social media and other digital platforms is a testimony to this rising infodemic. Fig. 1 depicts a sample rumour that emerged during the recent COVID-2019 impelled India Lockdown 1.0 regarding cut in pension disbursement of Indian citizens. The news was totally baseless and false and the Ministry of Finance, Government of India had to bust the fake news and give clarification.

Fig.1. Sample rumour and fact-checked in recent times

To ensure information credibility, many social networking sites such as Facebook, WhatsApp, Twitter and Instagram employ strategies and tools dedicated to identifying rumours and improving online accountability. These follow obligatory regulations or standard guidelines and rely on a combination of artificial intelligence, user reporting, and content moderators to implement rubrics for reliable and apposite

content filtering. But the strategies and code of practices are opaque to the users whereas the moderators are overwhelmed by the sheer volume of content and the ordeal that comes from sifting through vexing posts. Furthermore, the aggressive virality of rumours is an added nuisance [5]. Often, despite debunking a rumour, a re-posting of the same claim emerges. Thus, automated debunking of rumours and combating their viral spread is the need of the hour.

Typically, debunking and resolving a rumour entails four sub-tasks: detecting the rumour, tracking the rumour, classifying the stance and veracity [6]. An absolute rumour resolution framework involves integration and interaction of all these sub-tasks. Detecting rumours as a primary step may facilitate effective evaluation of the subsequent sub-tasks. Typically rumours in microblog posts are associated with a set of events and their lifecycle depends on their temporal characteristic [7]. The rumours can span over a longer period of time typifying an unrelenting and long-standing character or unfold in chorus to an occurrence of a breaking news event, which is with no history. Studies report a substantial interval between the first appearance of a rumour and its resolution [8]. The time period for determining the truth value of a breaking-news rumour may take three to eleven hours [Fig.2].

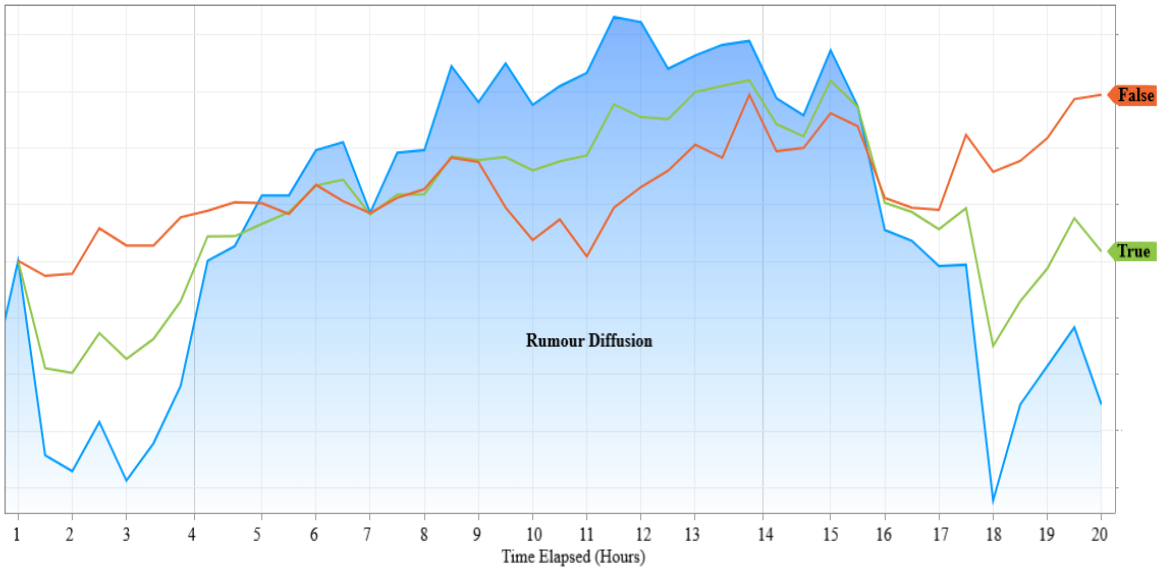

Fig.2. Breaking News Rumour Diffusion Model

Timely detection of the legitimacy of rumours is a strategic aspect in averting their viral spread [9]. An early stage debunking is necessary as the intensification is at its peak at the inception of an event. Computationally intelligent models with self-learning and generalization capabilities can facilitate automatic rumour detection. The use of different content, user and network based features have been reported in literature [1, 6]. Linguistic semiotic features can remarkably assist in identifying rumours during the initial propagation phase. These include vocabulary, structure, and grammar of oral/written language are the fundamental extractable textual features language which can contribute considerably as breaking news rumours are mostly circulated as trending stories and hashtags. It is thus important to automatically learn new, hidden features in natural language and their correlations from the input text itself for a real-time unfolding news story or event. Correspondingly, automatic rumour depends on feature set and learning model.

Rumour detection is quintessentially a text classification task which intends to categorize the incoming social media post as rumourous or not [10]. Feature engineering is a crucial sub-task in text classification, which is essential for the conversion of data into a machine learning-ready format. The primary problem pertaining to text-based rumour detection is the lower detection rate, wherein the classifiers fail to classify the misinformation accurately, which has a further derogatory effect on the detection rate and accuracy of

the system. Deep learning models have attained state-of-the-art results on many natural language processing tasks problems. Previous studies also confirm that feature selection allows faster training of machines and increases the accuracy of trained models, as few relevant features are better to train the model than huge amounts of irrelevant and redundant features [11]. It also benefits by avoiding the curse of dimensionality and overfitting, thus improving the accuracy and efficiency of the classifier. The feature selection techniques are broadly classified into filter methods, wrapper methods and embedded methods [12, 13]. Filter methods define selection techniques that rely on characteristics in the data itself by considering each feature individually and assessing its importance in prediction. On the other hand, wrapper techniques use a 'greedy' approach to generate an optimal feature set by assessing all possible feature combinations. Embedded methods define selection techniques that occur together while model fitting. While filter methods in general are fast and model agnostic, they tend not to select the optimal feature subset. Wrapper methods, by examining model performance on all (or several) feature subsets tend to find the best subspace for a given algorithm. But, as they build loads of models, they tend to be very computationally expensive, and often impracticable. Embedded methods capture the best of both worlds, and this is why they are very often the methods of choice.

This research seeks to include the pros of both deep learning to maximize utilization of unstructured data and feature selection techniques as a solution for such computational problems with the increased number of features, the training time surges rapidly and the risk of overfitting is increased too. A hybrid model for the rumour detection is proposed, where deep neural learning (Convolution neural network) is used in convergence with a filter-wrapper (Information gain –Ant colony) optimized Naive Bayes classifier enhanced accuracy in prediction results. CNNs are best at feature extraction with improved capabilities of representation and learning [14]. At the same time, naïve Bayes is an efficient classifier which is very simple to build and robust to outlier and irrelevant features. The proposed deep neural architecture consists of a convolution neural net (CNN) with four archetypical layers: embedding, convolution, activation and down-sampling (pooling) layers with a Bayesian classifier at the output layer. That is, the rumour classification in the output layer is done using a Naïve Bayes classifier. This classifier is trained using a combination of two sets of features, that is, the features which are learnt using the CNN and an optimized feature vector generated using the filter-wrapper technique, information gain (IG) [15] and ant colony optimization (ACO) [16]. This combined feature vector is used to train the Naïve Bayes classifier to predict the rumour. Thus, the proposed deep neural model, CNN-$_{IG\text{-}ACO}$NB has two primary design components:

- Firstly, instead of softmax regression which is normally used in the output layer of CNN, we use a Naïve Bayes classifier. Naïve Bayes is a generative model as compared to the discriminative softmax regression (logistic regression) model which is routinely used in the output layer of the CNN. Logistic regression tends to overfit if the training data is less. Alternatively, a naïve bayes classifier performs well even with less training data and has faster training time. This abets quick learning which is favorable in the case of rumour detection where early prediction can save streaming and virality.
- Secondly, as the Bayesian classifier can suffer from oversensitivity to redundant and/or irrelevant attributes and lead to a decline in the performance of the classifier. Feature selection may filter features leading to reduced dimensionality of the feature space. The selection of relevant feature subset is essential for both interpretability and predictive accuracy of the classifier. This research makes use of the hybrid feature selection technique, IG filter- swarm based ACO wrapper, reported by Bhatia and Sangwan [13] for detecting & predicting anomalies for IoT-based real-time abuse. This hybrid helps to select features with maximum relevance and minimal redundancy to train the prediction model.

Hence, the proposed CNN-$_{IG\text{-}ACO}$NB model, utilizes CNN as an automatic feature learner and NB as a rumour classifier. The Naïve Bayes classifier is trained by combining CNN generated feature vector with the IG-ACO optimized feature vector. The performance of the CNN-$_{IG\text{-}ACO}$NB model is validated on the PHEME rumour dataset [17]. A comparison is done against the state-of-the-art conditional random field (CRF) classifier [18]

2. Rumour: Taxonomy & Tasks

The newfound social media landscape for communication, disseminating information and voicing opinions brings to us substantial risks of fabricated information. Much of the discourse on 'online information fabrication' conflates three notions: misinformation (honest mistakes), disinformation (rumours, fake news and manipulated content) and mal-information (information leaks, harassment and hate speech), with each playing its part in contributing to the pollution of our information streams. These vary in accordance to the truth value of the content and the intent of information being created, produced or distributed (Fig.2). That is, dis-information encompasses absolute lies with no truth and is intentionally created to harm an individual, group, organization or country. Comparatively, misinformation is an erroneous mistake though the information is false, but it is not created with the intention of causing harm, rather it. Mal-information is grounded on reality but either taken entirely misrepresented, misquoted or manipulated, with malicious intent to inflict harm on a person, organization or country.

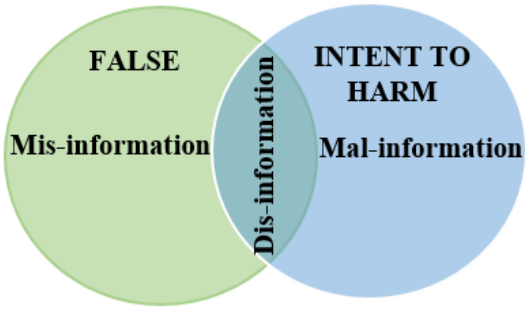

Fig.2. The 'information disorders' in social media

Undeniably, these 'information disorders' [19] that affect the social web have exposed us to the relentless virtual transgressions of lies, falsehoods and hate-crimes on the Web. The ease in online account creation, posting accessibility, broad latitude and virality makes social media an ideal and seamless choice for perpetrators as they tend to hide behind fake or hacked profiles to spread gossip or misleading stories. The economics of social media too favors rumours, hate-speech, pseudo-news, alternative facts or fake news [6, 20-21].

Simply put, a rumour is an assertion whose truth value is unverified. The rumour resolution process consists of four phases or subtasks as shown in Fig.3.

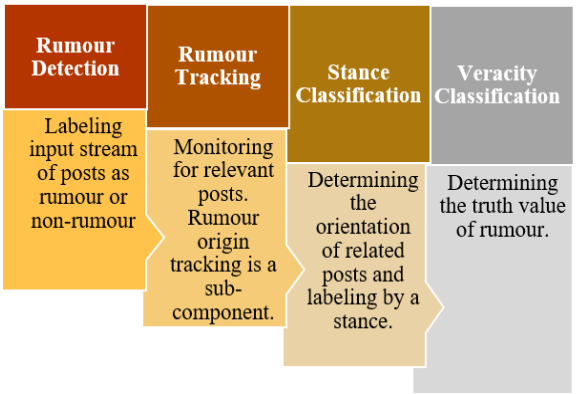

Fig. 3. The Rumour Resolution Pipeline

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

- *Rumour 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 rumourous 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.

3. Literature Review

Detecting rumours is essential, keeping in mind the volume and velocity of user-generated information on social media. Social media allows information propagation regardless of the source verification status and truth value. Forwarding and sharing content combined with the lack of validation fuels rumours as it permits exchange and broadcasting at an unmatched level. Nevertheless this can be harmful when users are exposed to damaging or undesirable content. Also, most social media platforms allow users to form groups based on their shared interests; however, such virtual alignments may lead to the creation of echo-chambers in which participants' own views are amplified and reinforced. Such echo-chambers also make unconfirmed posts appear more trustworthy. When a group member receives a certain piece of information, they might think that the information is truthful because it is from their "own" people.

Automatic rumour detection in social media data, especially on Twitter and Sina Weibo has been reported in various studies. In 2016, Zubaiga et al. [17] A comprehensive survey was given in 2018 by Zubiaga et al. [6]. The authors discussed the existing literature with respect to the various sub-tasks of rumour resolution. Various machine learning and deep learning models have been used to detect rumourous posts in microblogs. In 2018, Kumar and Sangwan [10] performed an analysis using a variety of ML for rumour detection. A range of features including text-based, user-based and network-based features have been used to train the learning models for detection and prediction of rumours [22-27]. Recently deep learning models have also been used for rumour detection in textual modality. RNN [28], attention-based RNN [29], hybrid of CNN with RNN [30] and LSTM with RNN [31] have reported superlative results. Multimodal rumour detection using LSTM and RNN with attention has also been proposed by Jin et al. [32]. The sequential classifier model, CRF, was given by Zubiaga et al. [18]. This research suggests building a hybrid learning model that seeks convergence of deep neural models and feature optimized machine learning techniques.

4. The Proposed CNN-$_{\text{IG-ACO}}$NB Model

The proposed hybrid of deep neural model and filter-wrapper feature selection entails the following components (Fig.4):

- CNN for automatic feature learning
- IG+ACO for optimized feature selection
- NB for rumour classification

The first component defines, initializes and trains the CNN which is seeded using the ELMo 5.5B embeddings [33]. ELMo generates the vectors for a word based on context. It is a character-based model using character convolutions and can handle out-of-vocabulary words. This fits the breaking news rumour type vector representation with words that are not seen in training. The textual features are converted into numerical data that can further be used for performing convolutions. The model uses three layer convolutional architecture with a total of 100 convolution filters each for window size (3, 3). The dropout regularization is set to 0.5 to ensure that that model does not over fit. The ReLU activation function is used

for introducing nonlinearities into the model which generates a rectified feature map. Max-pooling is used as a down-sampling strategy. The output layer in our model has a Naïve Bayes classifier. This classifier takes a concatenated feature set obtained by combining the learned vector representations from CNN (the output of top hidden layer) and a set of optimal features generated by applying IG+ACO on the training data set. Finally, the NB classifier categorizes the post as rumour or non-rumour.

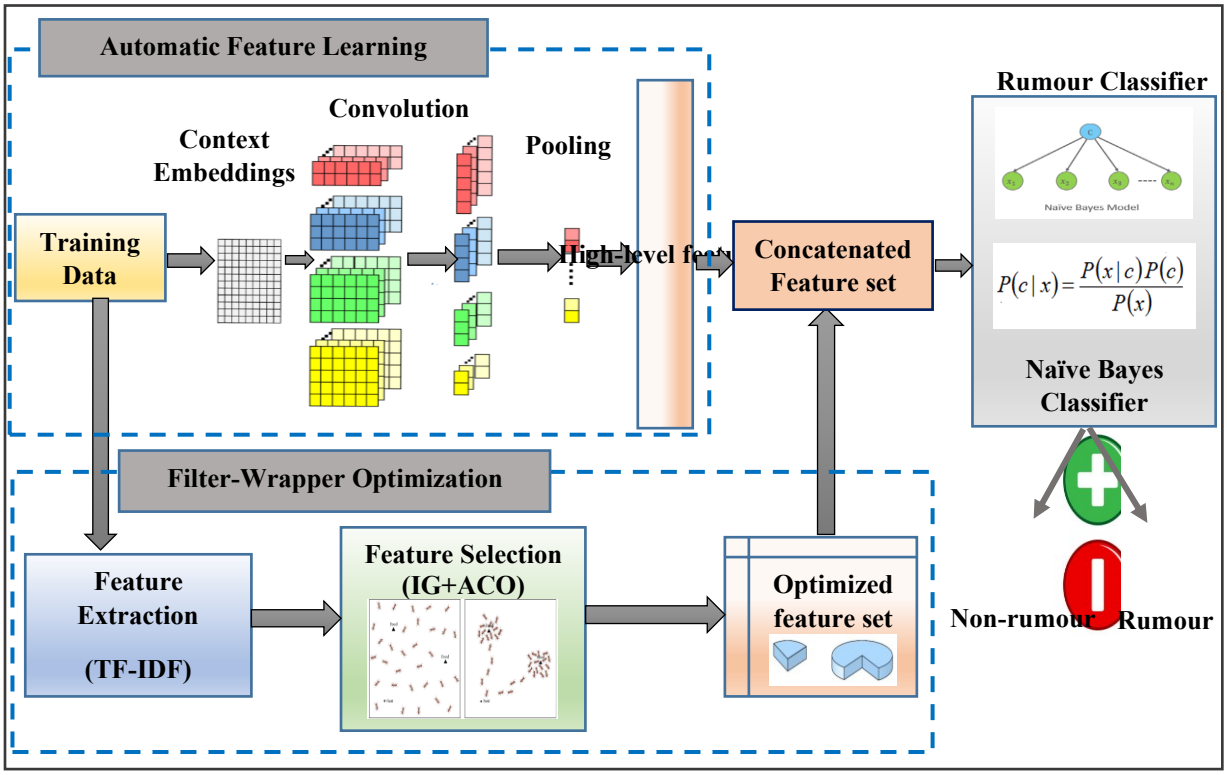

Fig.4. The architecture of proposed CNN-$_{IG\text{-}ACO}$NB

4.1. PHEME Dataset

The benchmark PHEME dataset used in this research for rumour detection [17] has tweets related to five breaking news events annotated for the 'rumour' and 'non-rumour' categories by expert journalists. These events are:

- *#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, Michael Brown Jr., an 18-year-old African American was fatally shot by a white police officer, 28-year-old Darren Wilson, in the city of 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 scheduled for the international Germanwings Flight 9525from Barcelona–El Prat Airport in Spain to Düsseldorf Airport in Germany crashed 100 kilometres (62 mi) north-west of Nice in the French Alps, killing 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 HaronMonis 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.

The label distribution for events within the dataset is shown in fig.5.

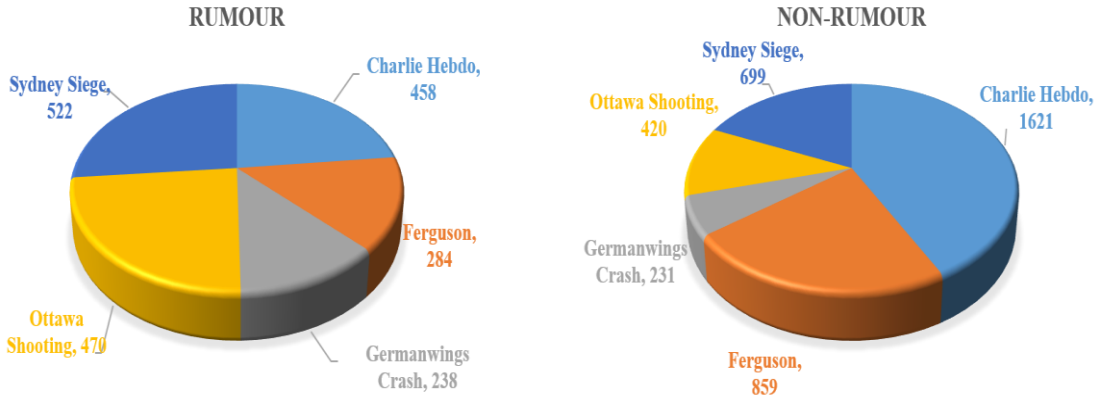

Fig.5. Label distribution for events in PHEME Dataset

4.2. Preprocessing

Preprocessing is the task of preparing the data in a manner, which is easier for the machine learning model to comprehend. The raw data is transfigured into clean data which is then used as input to the model.

4.3. Automatic Feature learning- Convolution Neural Network (CNN)

CNNs are usually apposite in computer vision tasks, however more recently their application to various NLP tasks have shown encouraging results [34]. Identical to the representation of images as an array of pixel values, text can also be represented as an array of vector where 1-dimensional convolutions are performed to pick up patterns in sequential data. CNNs are much faster to train, because of the batching. Typically, CNN architectures for text classification can either be character-level CNN or word-level CNN. In character-level CNN, input text is represented as $k*n$ matrix of one-hot encoding of the characters whereas in word-level CNN the input text is represented as $n*k$ matrix using word embeddings. In this research, word-level CNN is used. Fig. 6 depicts the architecture of a typical CNN Model.

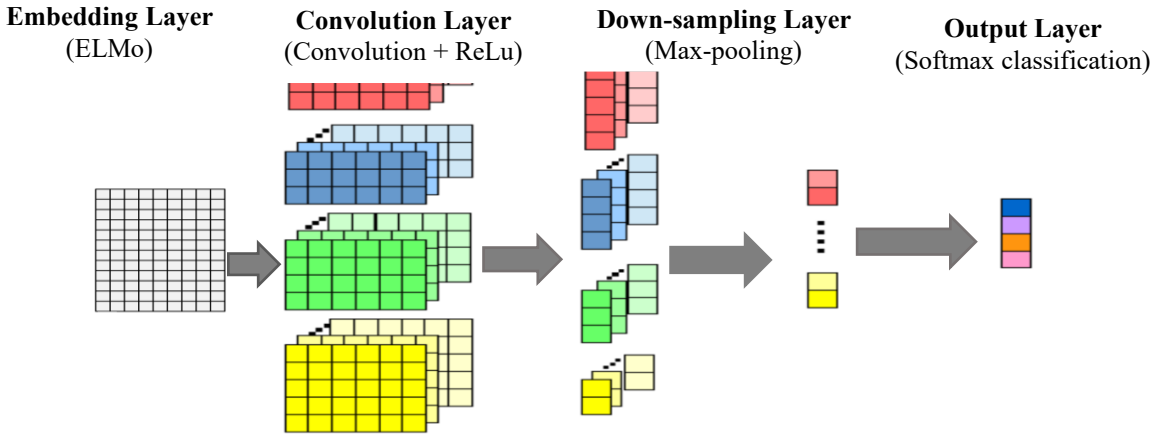

Fig.6. Typical CNN architecture

The posts from the dataset are extracted individually and are pre-processed to clean the posts, by removing the stop words and converting the words into their stems. These pre-processed posts are given as input to the embedding layer, where ELMo 5.5B model is used as the word vector learning technique to seed the classifier. Word vectors are used to represent the relationship across words, sentences, and documents. They are simply the vectors containing numbers that show and map word meanings for the model to understand. The vector representation of the text is provided to the filtration layer of the convolution having 128 filters of 8 size each, after applying a randomized function over the vectors. These filtered feature vectors are given as an input to a non-linear function that acts as a linear function and uses stochastic gradient descent to train the model and to avoid the saturation problem of other activation functions. The Rectified Linear Activation function is used in the third layer of CNN and so is named as the ReLU Activation layer. The ReLU function, f used in the model is

$$c_i = f\left(\sum_{j,k} w_{j,k}\left(X_{[i:i+h-1]}\right)_{j,k} + b\right) \qquad (1)$$

ReLU layer provides a half rectified feature map which is passed onto the fourth layer to perform downsampling by applying max-pool operation on the output feature vector matrix of ReLU layer over a feature matrix of 2×2, $c_{max} = \max(c)$. The max operation identifies the maximum value out of the sample feature map of 2×2 and converts the input feature matrix of shape 8×8×1 to 3×3. The output of the downsampling layer is a max-pooled feature map representation of the input tweet, which will provide a similar result for the minorly modified tweet. In a typical CNN that uses the fully connected output layer with softmax activation, this representation is used as an input to finally classify the tweet as positive (+1) of rumour or negative (-1) of rumour. But in the proposed model, the softmax output layer of the CNN is replaced by an easy to interpret and scalable Naïve Bayes classifier. NB learns a concatenated feature set generated by combining the high-level features obtained by CNN and the optimized feature set obtained using hybrid IG-ACO techniques to achieve the task of rumour classification as already shown in fig.5.

4.4. Feature Selection using Filter-Wrapper (IG-ACO)

In this work, to create the initial feature matrix, term frequency-inverse document frequency (TF-IDF) is used. TF-IDF is a weighting scheme for measuring the importance of a word with respect to a complete document [13]. It also checks the relevance of the keyword throughout the corpus. Feature selection techniques facilitate reduction in the number of input variables based on the usefulness to target prediction [1, 11, 13, 14]. Common categories of feature selection techniques include:

- Filter techniques attempt to assess the merits of attributes from the data, ignoring learning algorithm.
- Wrapper techniques the attributes subset selection is done using the learning algorithm as a black box.
- Embedded techniques performs automatic feature selection during training

Fig. 7 summarizes this hierarchy of feature selection techniques.

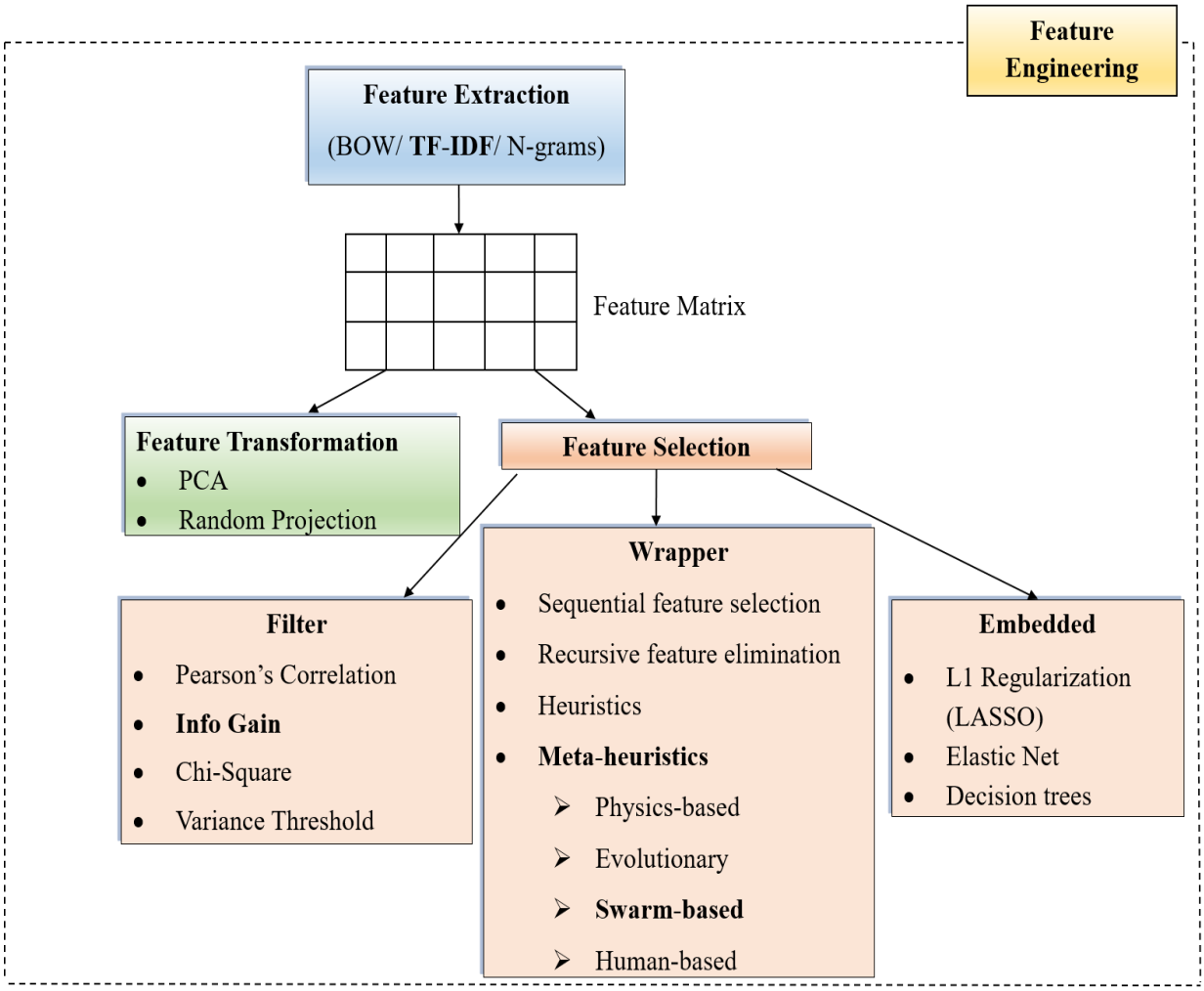

Fig. 7. Feature Engineering

In this work, feature selection is then done using the Information Gain *(*IG) filter and swarm-based (Ant-colony optimizer) ACO wrapper [13]. IG is calculated as:

$$IG(t) = -\sum_{i=1}^{m} p(ci)\log p(ci) + p(t)\sum_{i=1}^{m} p(ci|t)\log p(ci|t) + p(t')\sum_{i=1}^{m} p(ci|t'')\log p(ci|t'') \quad (2)$$

where, c_i indicates the i^{th} class; $p(c_i)$ indicates the probability of the i^{th} 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 conditional probabilities given the presence and absence of the feature t resectively.

In contrast to the filter methods, wrapper methods are based on the "usefulness" of features with respect to the classifier performance. Given by Dorigo [16] in 1992, ACO is inspired by the communication process used by ants. The algorithm for ACO is given as:

Algorithm 1: Ant Colony Optimization

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

The algorithm of the proposed CNN-$_{IG+ACO}$NB model is given next.

Algorithm 2: Hybrid Learning Model (CNN+$_{IG-ACO}$NB)

1. **Input:** *Train, Dev, Test, PHEME* Dataset **Output:** *Ac* 1: **Begin**: BuildNet()
2. **Initialize**: InitializeNet(Net)
3. **Repeat while** termination condition is satisfied **do**
4. error←TrainNet(Net, Train, Dev)
5. **End-while**
6. **Select**$Feature_{IG\text{-}ACO\_opt}$←IG-ACO(Train, Dev)
7. **Select**$Feature_{CNN\_rel}$←CNN(Train, Dev)
8. Hid_{Train}←GetTopHiddenLayer(Net, Test)
9. $Features_{concat}$←$Feature_{CNN\_rel}$ + $Feature_{IG\text{-}ACO\_opt}$
10. $Model_{NB}$←NB_{Train}($Features_{concat}$)
11. Hid_{Test}←GetTopHiddenLayer(Net, Test)
12. $Test_{concat}$←Hid_{Test} + $Feature_{opt}$
13. Ac←NB_{Test}($Model_{NB}$, $Test_{concat}$)
14. **return** (Ac)

4.5. Naïve Bayes Classifier

The Naive Bayes is a linear classifier using Bayes theorem and strong independence condition among features. Given a data set with n features represented by

$$F_1, F_2, F_3, \ldots, F_n \quad (3)$$

Naive Bayes states the probability of output: Y from features F_i is,

$$P(Y|F_1, F_2, F_3, \ldots, F_n) = P(Y|F_1)\,P(Y|F_2)\,P(Y|F_3)\ldots P(Y|F_n) = \prod_{i=1}^{n} P(Y|Fi) \quad (4)$$

This requires that the features F_i are conditionally independent. From Bayes theorem:

$$P(Y|F_i) = \frac{P(Fi|Y)\,P(Y)}{P(Fi)} \quad (5)$$

Softmax regression (or multinomial logistic regression) is a generalization of logistic regression which can handle multiple classes as compared to binary classes in the latter case. The learning mechanism is a bit different between the Naive Bayes (generative model) and Logistic regression (discriminative model).

- *Generative model:* Naive Bayes models the joint distribution of the feature X and target Y, and then predicts the posterior probability given as P(y|x)
- *Discriminative model:* Logistic regression directly models the posterior probability of P(y|x) by learning the input to output mapping by minimizing the error.

In 2001, Ng and Jordan [35] provided a mathematical proof of error properties of both logistic regression and naïve Bayes models. Their study concluded that when the training size reaches infinity the discriminative model: logistic regression performs better than the generative model Naive Bayes. However the generative model reaches its asymptotic faster (O (log n)) than the discriminative model (O (n)), i.e., the generative model (Naive Bayes) reaches the asymptotic solution for fewer training sets than the discriminative model (Logistic Regression). Naïve Bayes classifiers require a small amount of training data to estimate the necessary parameters and are extremely fast. As the size of the PHEME rumour dataset was small too, Naïve Bayes was a fitting choice.

5. Results and Discussion

The performance of the proposed model was evaluated for individual events within the PHEME dataset. The confusion matrix using the following values was computed for each event:

- *True Positives* (TP) - number of rumours correctly identified
- *False Positives* (FP) - number of non-rumours that were incorrectly identified as rumours
- *False Negatives* (FN) - number of rumours that were incorrectly identified as non-rumours
- *True Negatives* (TN) - number of non-rumours correctly identified

The confusion matrices for each individual event are shown in fig.8 with actual class on the horizontal axis and predicted class on the vertical axis.

Fig.8. Confusion matrix for individual events in PHEME

Two events, Charlie Hebdo and Ferguson unrest suffer from the class imbalance problem. Data skewness or class imbalance proves to be a major limitation in a classification task. In the case of class imbalance, the accuracy score is not used as an evaluation metric as it often leads to incorrect interpretation of performance. Thus, we have used F1 Score, Precision, and Recall as evaluation metrics to correctly represent performance on the PHEME dataset. We evaluate our model for individual events over five iterations with a leave-one-event-out approach [36]. Other than precision, recall, and F1 score, we have also

used AUC-ROC curves to judge the performance of our model. The AUC scores of the five events present in the PHEME dataset range from 0.75 to 0.90. Fig.9 shows the ROC curve for each individual event.

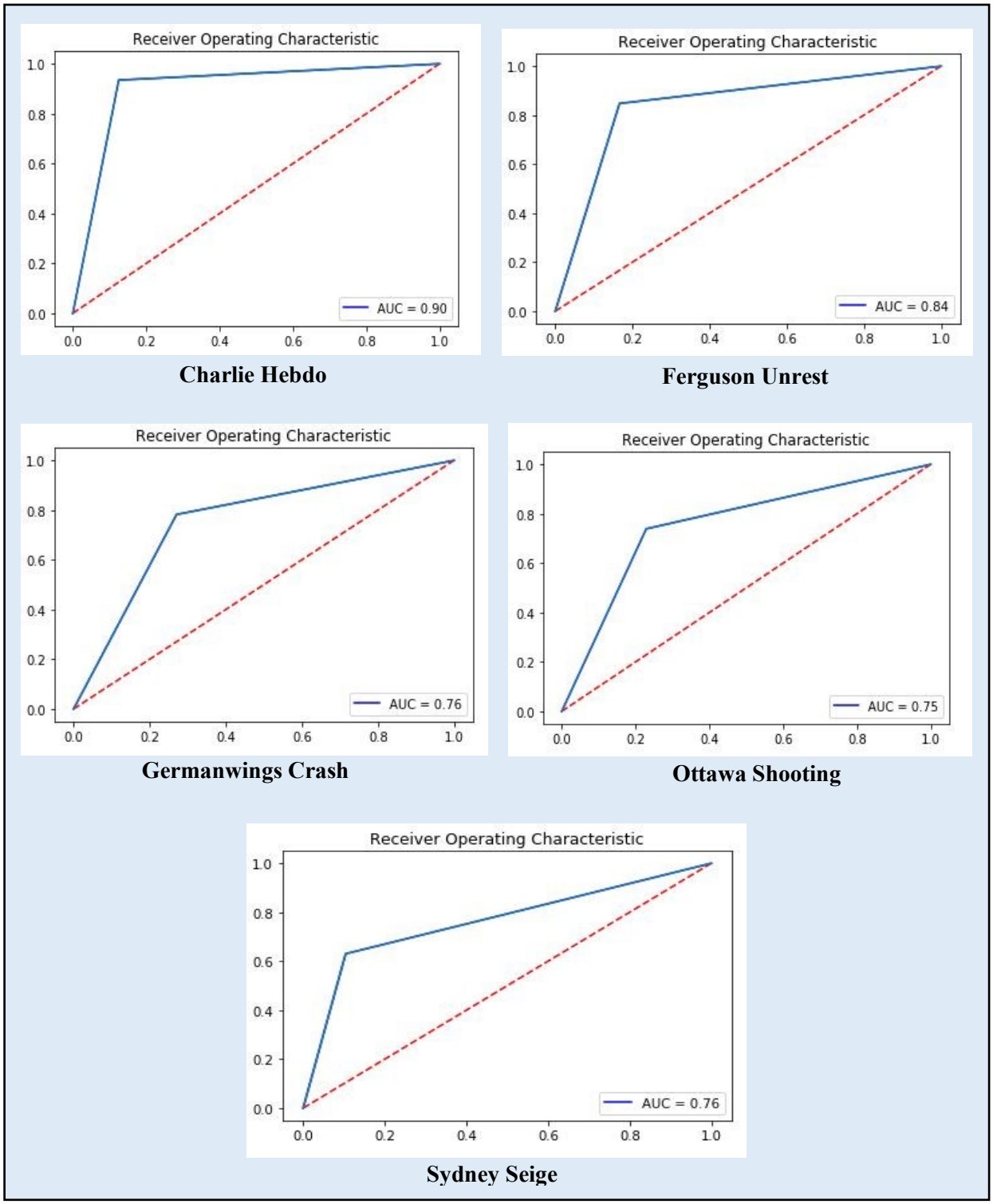

Fig.9. AUC-ROC for individual events in PHEME

The experiments show that our proposed model, which is a hybrid network, is able to handle multiple input types using the distinct classifiers to maximize the potential of each feature type. It achieves significant improvements over traditional methods of binary text classification. Table 1 depicts the results for each individual event in PHEME evaluated for the proposed CNN+$_{IG\text{-}ACO}$NB model and the state-of-the-art CRF Classifier [18]. The proposed model achieves higher precision compared to the state-of-the-art classifier for three events out of the five, the exceptions being Ottawa Shooting and Sydney Siege events. The recall scores achieved by the proposed model are also superior for all the five events. The model also manages to strike an equilibrium between recall and precision, a qualitative improvement over the state-of-the-art.

Table 1. Performance of CNN+$_{IG\text{-}ACO}$NB on individual events in PHEME

	Proposed CNN+$_{IG\text{-}ACO}$NB model			**CRF [18] State-of-the-art**		
Events	Precision	Recall	F1	Precision	Recall	F1
Germanwings Crash	**0.767**	**0.761**	**0.763**	0.743	0.668	0.704
Charlie Hebdo	**0.856**	**0.841**	**0.848**	0.545	0.762	0.636
Ottawa Shooting	0.749	**0.801**	**0.773**	**0.841**	0.585	0.690
Sydney Siege	0.740	**0.699**	**0.719**	**0.764**	0.385	0.512
Ferguson Unrest	**0.874**	**0.841**	**0.857**	0.566	0.394	0.465

Also, quite clearly, the proposed model does not let the events with class imbalance impede the classifier performance which is another improvement over the CRF classifier. CRF has an F1 score of 0.636 and 0.465 for Charlie Hebdo and Ferguson Unrest events respectively, whereas the proposed model yields a score of 0.848 and 0.857 for the same events. Same trend was observed for the remaining events which were free from the class imbalance problem, i.e., the proposed model outperforms the CRF for the Germanwings crash, Sydney Siege, and Ottawa Shooting events in terms of F1 scores. It is worth mentioning that the CNN+$_{IG\text{-}ACO}$NB model performs exceptionally well in terms of recall score and achieves an overall superiority over the CRF classifier.

The overall effectiveness of the CNN+$_{IG\text{-}ACO}$NB classifier is shown in table 2

Table 2. Classifier performance of the CNN+$_{IG\text{-}ACO}$NB classifier

Classifier	**P**	**R**	**F1**
CRF [16]	0.667	0.556	0.607
CNN+$_{IG\text{-}ACO}$NB [Proposed Model]	**0.776**	**0.745**	**0.732**

The results of using the hybrid IG-ACO technique were also evaluated on three different classifiers, namely, random forest (RF) and Naïve Bayes (NB), decision tree (DT) for all the five events of the PHEME dataset. We also compared the IG-ACO with the IG-Cuckoo search algorithm and trained the dataset on three different classifiers (NB, RF and DT). The results are given in table 3.

Table 3. Performance with TF-IDF + IG filter+ 2 Wrappers (ACO/ Cuckoo)

Filter-Wrapper	**Event**	**NB**			**RF**			**DT**		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
IG +ACO	Germanwings Crash	0.67	0.67	0.67	0.64	0.58	0.54	0.60	0.57	0.55
	Charlie Hebdo	0.79	0.78	0.69	0.79	0.79	0.70	0.79	0.78	0.72
	Ottawa Shooting	0.67	0.62	0.60	0.68	0.67	0.67	0.75	0.54	0.39
	Sydney Siege	0.64	0.59	0.47	0.72	0.59	0.45	0.70	0.58	0.43
	Ferguson Unrest	0.76	0.76	0.66	0.79	0.76	0.66	0.82	0.76	0.66
IG + Cuckoo	Germanwings Crash	0.65	0.63	0.62	0.73	0.58	0.50	0.60	0.57	0.55
	Charlie Hebdo	0.75	0.78	0.69	0.79	0.79	0.70	0.78	0.78	0.72
	Ottawa Shooting	0.70	0.69	0.68	0.76	0.60	0.54	0.75	0.54	0.39
	Sydney Siege	0.58	0.58	0.47	0.72	0.59	0.45	0.69	0.58	0.43
	Ferguson Unrest	0.71	0.75	0.65	0.79	0.76	0.66	0.82	0.76	0.66

Though the best average results over the entire dataset were given by TF-IDF + IG + Cuckoo but IG+ACO performed better with the NB classifier which was decided to be used as the output classifier in the proposed model. A reduction of 77% in the feature set was observed using IG+ACO which was comparable to the 79% reduction in features observed using IG-Cuckoo as shown in fig. 10. For both TF-IDF+IG+ACO NB and TF-IDF+IG+Cuckoo, NB took 0.30 seconds for model building whereas RF took maximum time (10 times more than NB).

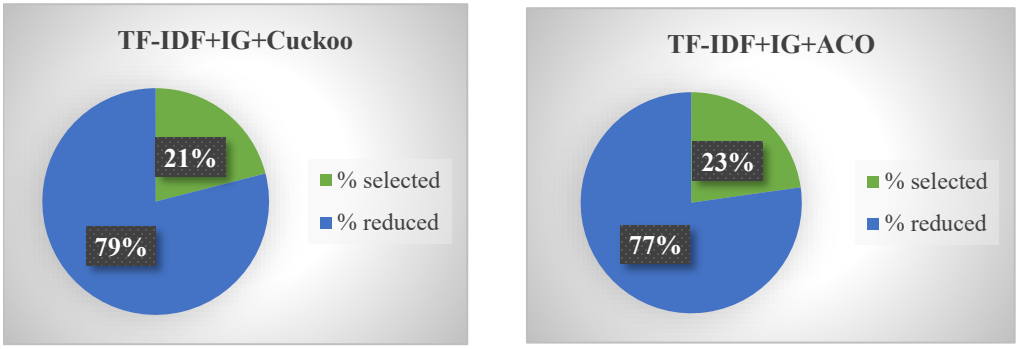

Fig. 10. Feature Reduction using IG and ACO/Cuckoo wrappers

6. Conclusion

Rumours proliferate in times of crisis. The uncertainty and significance of the situation, combined with the lack of information fuels rumours in the virtual social world. It is thus imperative to question the tangibility of information. As a solution to debunk online rumours, this study proffered a novel model for real-time rumour classification which learns combined features from the high level features from CNN and the optimized features obtained using hybrid information gain filter-meta-heuristic ant colony optimization wrapper feature selection technique. The classifier is an easy to interpret Naïve Bayes classifier that replaces the final logistic regression (softmax) layer in the CNN architecture to classify tweets into binary categories of rumour and non-rumour. The model is evaluated on the PHEME benchmark dataset and compared with the existing state-of-the-art. Results validate superior F1 of 0.732 using the proposed CNN+$_{\text{IG-ACO}}$NB rumour classifier.

Our approach uses only the text-based features whereas meta-features such as re-tweet count and user-based features can be learned separately to build a robust model to uncover rumours. Further, the use of country-specific content written in native language is also compounding the linguistic issues in detecting rumours. As a future direction, next stages of the rumour resolution pipeline can be explored using the hybrid model. Also, as this work presents text-based rumour detection, context modelling can be done to improve the detection and debunking of rumourous stories. Further it is also imperative to leverage information from different media platforms and different languages to verify rumor automatically and analyze multimedia rumor verification datasets, such as CCMR and MediaEval 2015's Verifying Multimedia Use (VMU 2015).

References

1. Kumar, A., Sangwan, S. R., & Nayyar, A. (2019). Rumour veracity detection on twitter using particle swarm optimized shallow classifiers. *Multimedia Tools and Applications*, *78*(17), 24083-24101.
2. Tripathi, A. K., Sharma, K., Bala, M., Kumar, A., Menon, V. G., & Bashir, A. K. (2020). A Parallel Military Dog based Algorithm for Clustering Big data in Cognitive Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*.
3. Sangwan, S. R., & Bhatia, M. P. S. (2020). D-BullyRumbler: a safety rumble strip to resolve online denigration bullying using a hybrid filter-wrapper approach. *Multimedia Systems*, 1-17.
4. Bhatia, M. P. S., & Sangwan, S. R. (2020). Debunking Online Reputation Rumours Using Hybrid of Lexicon-Based and Machine Learning Techniques. In *Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019)* (pp. 317-327). Springer, Singapore.

5. Tellis, G. J., MacInnis, D. J., Tirunillai, S., & Zhang, Y. (2019). What drives virality (sharing) of online digital content? The critical role of information, emotion, and brand prominence. *Journal of Marketing*, *83*(4), 1-20.
6. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, *51*(2), 1-36.
7. Zubiaga, A., Liakata, M. Procter, R. (2016). Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv:1610.07363*
8. Vosoughi, S., Roy, D. and Aral, S., 2018. The spread of true and false news online. *Science*, *359*(6380), pp.1146-1151.
9. Zhao, Z., Resnick, P. and 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.
10. Kumar, A., & Sangwan, S. R. (2019). Rumor detection using machine learning techniques on social media. In *International Conference on Innovative Computing and Communications* (pp. 213-221). Springer, Singapore.
11. Sangwan, S. R., & Bhatia, M. P. S. (2020). Denigration Bullying Resolution using Wolf Search Optimized Online Reputation Rumour Detection. *Procedia Computer Science*, *173*, 305-314.
12. Ebrahimpour, M. K., & Eftekhari, M. (2017). Ensemble of feature selection methods: A hesitant fuzzy sets approach. *Applied Soft Computing*, *50*, 300-312.
13. Bhatia, M. P. S., & Sangwan, S. R. (2021). Soft computing for anomaly detection and prediction to mitigate IoT-based real-time abuse. *Personal and Ubiquitous Computing*, 1-11.
14. Kumar, A., & Jaiswal, A. (2020). A Deep Swarm-Optimized Model for leveraging Industrial Data Analytics in Cognitive Manufacturing. *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.3005532.
15. 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.
16. Dorigo, M. (1992). Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*.
17. Zubiaga, A., Wong Sak Hoi, G., Liakata, M., & Procter, R. (2016). PHEME dataset of rumours and non-rumours. figshare. *Dataset. doi*, *10*, m9. Available from: https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619/1
18. Zubiaga, A., Liakata, M., & Procter, R. (2017, September). Exploiting context for rumour detection in social media. In *International Conference on Social Informatics* (pp. 109-123). Springer, Cham.
19. Bounegru L, Gray J, Venturini T, Mauri M. A Field Guide to 'Fake News' and Other Information Disorders. A Field Guide to" Fake News" and Other Information Disorders: A Collection of Recipes for Those Who Love to Cook with Digital Methods, *Public Data Lab*, Amsterdam (2018). 2018.
20. Li, G., Dong, M., Yang, F., Zeng, J., Yuan, J., Jin, C., & Zheng, B. (2020). Misinformation-oriented expert finding in social networks. World Wide Web, 23(2), 693-714.
21. Zhang, P., Bao, Z., Niu, Y., Zhang, Y., Mo, S., Geng, F., & Peng, Z. (2019). Proactive rumor control in online networks. *World Wide Web*, 22(4), 1799-1818.
22. Cao, J., Guo, J., Li, X., Jin, Z., Guo, H., & Li, J. (2018). Automatic rumor detection on microblogs: A survey. arXiv preprint arXiv:1807.03505.
23. Takahashi, T., & Igata, N. (2012, November). Rumor detection on twitter. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems* (pp. 452-457). IEEE.
24. 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* (pp. 1-7).
25. 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).
26. Liu, Y., & Xu, S. (2016). Detecting rumors through modeling information propagation networks in a social media environment. *IEEE Transactions on computational social systems*, *3*(2), 46-62.
27. 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.
28. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. F., & Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks.
29. 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.

30. 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.
31. Alkhodair, S. A., Ding, S. H., Fung, B. C., & Liu, J. (2020). Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management*, *57*(2), 102018.
32. 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).
33. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
34. Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, *6*, 23253-23260.
35. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848).
36. Kumar, A., Shrivastava. A. (2020). Rumour Detection in Benchmark Dataset using Attention-Based Residual Networks. *International Journal of Advanced Science and Technology*, *29*(3), 14682 -. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/31956