Ensemble based Classification of Dynamic Rumor Detection in Social Networks for Green Communication

1R.Amutha, 2D.Vimal Kumar

1Research Scholar, Department of Computer Science, Nehru Arts and Science College, Coimbatore, TamilNadu, India. E-mail: amutharajan@gmail.com
2Associate Professor, Department of Computer Science, Nehru Arts and Science College, Coimbatore, TamilNadu, India. E-mail: drvimalcs@gmail.com

Abstract

Microblogging is the one that gained more attention among social media users. In business and individuals adopting green communication technology reduces the power consumption and reducing the operational cost. Human beings spend very little time in reading such micro-blogs and they can quickly learn and understand them very easily. It is very popular among social media users. One of the known examples of a micro-blog is Twitter. Twitter is a micro-blogging site that allows users to communicate with others using texts known as tweets. Nowadays rumor is spreading very exponentially and it is impacting society in many ways. So detecting the rumor in twitter data is one of the most concentrated research issues in recent days. Those researches introduced different research techniques to handle rumors. However, those research techniques tend to have issues such as identifying effective ways for obstructing malicious rumors and difficulty in minimizing the negative influences. So need for a classifier that will give greater accuracy to solve this problem. Improved Adaboost using Probabilistic Weight Updation and Pruned Stacking algorithm are proposed in this paper to attain a high prediction rate of rumor detection. Proposed algorithms and AdaBoost are investigated using a publicly available PHEME
The experimentation results prove that proposed algorithms outperform existing algorithms in all the measures. Experimentation results also prove that all the algorithms produce better results when the class imbalance is handled using SMOTE.

**Keywords:** Class Imbalance, SMOTE, Semi Supervised Clustering, PHEME Data.

1 **Introduction**

Social media has gained more attention and popularity among many people irrespective of age because here we get connected not only with family, friends, and colleagues but also with other people who we have not met in real-time. Social media is not only used for communication purposes but also for creating business connections, advertising, marketing, etc. Especially social media helps in making news popular and to reach a wider population of an audience. So any information can easily reach a larger group of people. Every news organization is updating news on social media and also journalists post news on social media. So now reaching for news becomes quite easy. Social media is the one with a diversity of information [1]. There are possibilities for fake news or rumor to spread about someone because many users of social media are interested in getting likes for posts either in text or picture form and to gain more followers [2]. The rumors spread in social media are difficult to analyze and know the right piece of information because of the unauthorized person spreading rumors but he would be famous among other fake news spreaders. News organizations later try to put the right information instead of a fake one. They put the correct news and save an individual and society from believing fake news or rumors. According to a study it was found that almost 1/3rdof micro-blogs and social media news carried out in China are a fake piece of information. The following are certain rumor information carried out related to the economic condition of a country, political events, and social stability. The incident took place on 30th May 2018 that Deadly Nipah virus can spread to human beings by broiler chickens. Because of fake information, many people stopped consuming chicken and many dealers from Tamil Nadu also stopped buying chickens from Kerala and faced huge financial loss [3]. Another incident took place on 4thJanuary 2018, that was a video message containing information about a pillar collapse of a Bangalore Metro and this information created more panic situation among Bangalore residents who are using the Metro for transport. And even many news channels carried out this fake piece of information and as news breaking but this incidence was clarified with Bangalore Metro Rail Corporation Limited (BMRCL) stating it as a piece of fake news and requested news channels and social media users to stop spreading fake news among public and creating a panic situation [4].
Fake news is a kind of information which are novel news that spreads faster than a true piece of information. This is because many people tend to share news either knowingly or unknowingly [5]. The spread of fake news or rumor is having a serious negative effect among individual or society which are stated as follows: Rumors can make the audience easily question news organizations or channels carrying news and any news from social media even if it is true it’s very hard to believe. Fake news makes people not to believe real news. Fake news about a political party can impact its effect during an election poll [6].

Detecting fake news in social media gained more importance among several researchers because a fake piece of information each spoil the academic work, rumors are capable of spreading misinformation to people who are believing it [7]. The spread of fake news must be stopped otherwise it may cause ill-effect in society among people. The effect of fake news in society lay to be very serious and dreadful. Detecting fake news in social media must be done in a short span of time. ML (Machine Learning) has gained popularity for detecting fake news in social media [8]. Abulaish et al., in [9] machine learning trained classification model is applied for predicting news data sample whether it a fake (rumor) news or original (non-rumor) news.

The spread of rumor news has a huge negative impact on society, especially more fake news are carried out related to a political party, public opinion. So fake news must be detected and stop it from spreading as soon as possible. The research paper aims at addressing the problem of detecting fake or rumor news with rumor influence minimization.

This research paper is structured in order as Section 2 carries information about several Review of Literature, Section 3 carries information about the proposed algorithm of the study and research methodology, Section 4 carries information about result and discussion of this study, and Section 5 is about the conclusion and future research work on this topic.

2 Related Works

Thakur et al., proposed a framework for detecting rumor utilizing naïve Bayes classifier. This method was used as the process by implementing software and testing the data set of Facebook news posts by users. This data set for the research was taken from 3 large Facebook page and 3 mainstream political news site such as CNN, ABC News, and Politico. The outcome of the research was found to be with an accuracy of 70 percent in classification. The classification accuracy for a few pieces of news was very poor. This is due to the skewness of the dataset [10]. Li, Zhang, Si, et al., proposed a framework concentrating on the United States presidential Election of 2016, which was an important topic of discussion on Twitter. In this research, an analysis of rumor tweets was done with the followers of Presidential
Candidates Donald Trump and Hillary Clinton. The rumored tweet was detected by matching tweets related to the president’s election with verified rumor articles. For classifying various kinds of matching techniques were used such as Doc2Vec, Word2Vec, BM25, and TF-IDF. Through this approach, the outcome of the analysis yielded better results [11]. Ghanem et al., proposed a framework for rumor detection by automatic detection method with the combination of implicit features and shallow features of the text messages [12]. In this several approaches have been utilized such as Random Forest and SVM (Support Vector Machine). These approaches yielded better results. Ma et al., proposed a framework for automatic detection of rumors on Twitter and the process of identification of sources for rumor spread [13]. For this, the selected topic for rumor detection was the London Riots of 2011, and certain rumored and non-rumored tweets were used. For classification Weka Tool was used. The algorithm used in this research was J48. And an algorithm was proposed for finding the source for a rumored tweet.

Pathak et al., proposed a framework for investigating a rumor that was created in the past 2009 and detecting it through machine learning algorithms the KNN, and Naïve Bayes classifier to detect rumor tweets spreading in Twitter [14]. With the use of the Naïve Bayes classifier, the detection of tweets improved and obtained an accuracy of 85 percent for each set of rumors denied, rumor supported, and 75 percent for the queried rumor. Sicilia et al., proposed a framework using Multi-stage Reliability analysis [15]. Through which the rumor on Twitter is identified. Naïve Bayes is used for classification and the model is improved by giving more importance to the user features which helps in improving the accuracy of classification. The classifier is done by examining a thousand tweets that are unique from 700 Twitter Accounts. Li, Zhang, & Si, proposed a framework using an ultra-design known as the Tweet cred which is used to group the message tweets based on their credibility. 45 features are used for determining the credibility of each tweet [16].

Conforti et al., proposed a framework utilizing DRIMUX [17]. This model can be utilized for global popularity as well as for individual attraction of rumors related to the real situation. This research focus to solve the problems faced while detecting rumors utilizing dynamic rumor minimization with a novel classifier approach for text in tweets. Enayet and El-Beltagy [18] have used Linear SVC for SemEval Rumour Eval dataset. And authors concluded that Linear Support Vector Classification was the best algorithm to get the good results in terms of accuracy. Ma et al., [19] used Linear Support Vector Machine, SVM-Time Series, Decision Tree using Ranking method, Random Forest and Recurrent Neural Network Algorithms to handle Twitter rumour data set. Word embedding and Bag-of-words (BoW) were used as feature set. Authors finally come to the conclusion that BoW performed well compared to word embedding. Wu et al., [20] considered rumour data set with 100 reposts. Authors proposed SVM with a hybrid kernel technique to handle rumour data set. The
proposed hybrid approaches shown greater accuracy than other algorithms.

Liu et al., [21] used their own rumour data set. Algorithms such as Support Vector Machine with the radial basis function kernel decision tree; Naïve Bayes was used to measure the performance. Support Vector Machine gave the best results. Ma et al., [22] Twitter and SinaWeibo Decision Tree, Random Forest, Support Vector Machine with the radial basis function kernel. The proposed approach Support Vector Machine had produced the best accuracy of around 84.6%. Chang et al., [23] proposed new clustering approach to handle Twitter rumour dataset. Authors had applied clustering heuristic and based on that they categorized the rumour data sets as false or true rumour clusters. The results of the new clustering shown improved performances compared to other approaches.

Hamidian and Diab [24] applied supervised machine learning for tracking and stance classification of rumours using the dataset created by Qazvinian et al., [25]. Authors used J48 decision tree for classification. Performance was evaluated using two ways: 1. Six-way classification task (4 classes of stance and unrelated to rumour, four and not determined), 2. Three way classification task (Related to rumour, not determined, unrelated to rumour,) and then 4-class rumour stance classification. The second approach achieved better results compared to first approach. Zeng et al., [26] introduced new set of features based on LIWC (Linguistic Inquiry and Word Count) dictionaries. LIWC dictionaries were introduced by Tausczik and Pennebaker [27]. Authors investigated logistic regression, naïve Bayes, and random forest (RF) to classify the rumour stance. Authors considered only two classes, namely affirm and deny. RF achieved best results in terms of accuracy. Hybrid Ensemble with novel weighting for class labels and manually created rules were the new way of handling data to improve the accuracy of the algorithms (Wang et al., [28] García Lozano et al., [29] Srivastava et al., [30])

3 Methodology

This research focus to solve the problems faced while detecting rumors utilizing dynamic rumor minimization with a novel classifier approach for text in tweets. The proposed methodology of this research is shown in fig 1 as follows:

PHHEME Dataset has been considered in this research work. Preprocessing involves tokenization, stemming and stop word removal. Then SMOTE is appealed to conduct class imbalanced data. SSC is used to cluster with features. Then proposed algorithms are applied to detect rumour.
3.1 Social Network

Social Network can be formulated in terms of mathematics through Graph where \( G = (V, E) \) with a group of joints as \( V \) denotes who is using this network and directed to represented the edges as \( E \) which is used for representing the relationship of users.

3.2 Dataset

PHEME is a dataset popularly used for rumor detection dataset. It accommodates set of Twitter gossip and facts that posted through breaking news. The group of information is used for analyzing the performance of the proposed algorithm.

3.3 Dataset Preprocessing

For classifying the collected tweets the PHEME dataset is used and also for filtering the collected data. From filtering many redundant elements and inconsistent can be avoided. During the training phase if there are lots of redundant functions or noise then in such case, it will be very difficult in detecting the rumor news. Tasks performed for data pre-processing are discussed as below:
3.3.1 Emoticons

All the emoticons present in the data are replaced with corresponding word equivalents.

3.3.2 Lowercase Conversion

All the uppercase letters are converted into lowercase letters. This is needed for effective text classification.

3.3.3 Remove White Spaces, Punctuations

This process is done through using usual expression by identifying the URL and removing the URL from the tweet. The hashtag is removed using the symbol removal process. All the special characters used are removed completely. Then all the punctuations, accent marks, white spaces and other diacritics are removed from data.

3.3.4 Tokenization

This process is used for constructing the words into bag-of-words by which the words are splitted.

3.3.5 Removal of Stop-words

When the same word is repeated multiple times it is useless and will not be considered for classification. The repeated word can be reaped in this method. This will remove the repeated word without reducing the meaning of the text or a sentence.

3.3.6 Stemming

It is used for removing the suffix present in the word based on grammatical rules. In this work the Snowball stemmer is used.

3.4 SMOTE

SMOTE generates synthetic samples by taking each minority class illustration and preface for manufactured examples accompanied the together fragments of the j minority group which is close to neighbors. The method of power dominant the small groups of the forces to act like causal. Unfortunately, this technique doesn’t work well with text data because the numerical vectors that are created from the text are very high dimensional.
3.5 Semi-Supervised Clustering Algorithm (SSCA)

The anticipated work utilizes a novel Semi-Supervised Clustering Algorithm (SSCA) that considers rumor problem into search problem over feasible user’s identification. The proposed work grouping technique is developed to find the answer for gossip circulation problem by enhancing half Supervised Clustering Algorithm (SSCA). An ultimate gossip circulation results are captured by evaluating the similarity between rumor and normal tweets [31]. SSCA first groups similar words by identifying synonyms and relevant meaning. Then it forms cluster based on similarity of word meaning. Semi-supervised clustering uses pairwise constraint information to direct clustering process. This algorithm is used as a rumor propagation process model in social networks.

3.6 PSO_Ada Boost

In our previous work [32-33], PSO-Adaboost algorithm has been proposed to detect rumors. Particle Swarm Optimization (PSO) and AdaBoost algorithms are integrated in this work. Optimum weights are allocated in AdaBoost using PSO. AdaBoost Ensemble have the feature of combining many weak classifiers into one best strong classifier. Weights of the weak classifiers are determined in the each iteration. This results in larger weights for useless weak classifiers. It also leads to increase in system overhead. In order to solve this issue, PSO_AdaBoost uses the PSO to optimize the weights of the AdaBoost.

3.7 Pruned Stacking

Pruned Stacking combines the concept of pruning models and stacking. Overall diagram of pruned stacking is shown in fig.2. This project presented two-stage pruning method based on certain rules to improve the accuracy. The proposed pruned stacking is composed of pruning procedures with stacking.

3.7.1 Working Principle of Pruned Stacking

- Layer 1 (1st Level Learner): Train the individual classifier models on training data. Before passing these models to Layer 2, prune the unwanted models (Model having less than 80% accuracy)
- Layer 2 (2nd Level Learner): Combine all individual classifier models and generate Meta model for each classifier.
- Layer 3 (3rd Level Learner): Combine all Meta models and we will have stacked model for training data.
- Finally, predict using the stacked model that has been made for testing data.
3.7.2 Pseudo code – Pruned Stacking

- $x_i$ -> Observations, $y_i$ -> Set of labels $M$ -> Models
- $D$ -> Dataset
- Input: $D \{ (y_i, x_i) | y_i \in c, x_i \in X \}$
- Output: An ensemble classifier $H$

Step 1: Process first; proceed high – low grouping
Step 2: For $d \geq 2$ to $E$ do
Step 3: Acquire a least categorize $h_t$ depend on $E$
Step 4: Calculate the accuracy for each first-level classifier
Step 5: Eliminate first level classifier models which is having accuracy < 80
Step 6: Process 2: Construct new model
Step 7: For $t \geq 1$ to $R$ do

---

**Figure 2** Pruned Stacking
Step 8: Construct a new model that contains \((x)\) for \(j = 1\) to \(T\)\}

Step 9: Process 3: Learn a next aligned grouping.

Step 10: Learn a latest grouping \(h\text{new}\) depend on the latest constructed model.

Step 11: Return \(H(y) = h\text{new} (h1(y), h2(x1) \ldots hT (x))\)

3.8 Improved Adaboost using Probabilistic Weight Updation

In the standard AdaBoost, the weight given to strong classifier depended on the overall error of the each classifier. Having 2 low categorize with the similar mistake, their belief provided same density. Even the possibility of differentiate the plus and minus classes might vary highly. In order to solve this issue, this paper introduced a latest density machine. The density were proceed to be strong classifier considers both positive and negative classes. Therefore, this new weight system is used to construct a strong classifier that decreases the FPR. In standard Adaboost, misclassified records weight updation is based on total number of misclassified records divided by total number of records. In our proposed algorithm, misclassified records weight updation is done based on below approach.

- **EPFP** (Error Probability of False negative) = True Negative/Group of records
- **EPFN** (Error Probability of False Positive) = True Positive/Group of records
- Misclassified records weight updation for False Positive= Error Probability of False Positive
- Misclassified records weight updation for False Negative= Error Probability of False Negative

Weight Updation in Standard Adaboost for both False Positive & False Negative.

\[
\beta = \log \left( \frac{ET}{(1 - \varepsilon t)} \right) \tag{1}
\]

In proposed AdaBoost algorithm, Weight Updation in our Adaboost for False Negative Records,

\[
\beta = \log \left( \frac{\varepsilon t}{(1 - \varepsilon t)} \right) \ast \left( \frac{1 - \text{EPFN}}{\text{FN}} \right) \tag{2}
\]

Weight Updation in our Adaboost for False Positive Records,

\[
\beta = \left( \frac{\varepsilon t}{(1 - \varepsilon t)} \right) \ast \left( \frac{1 - \text{EPFP}}{\text{FP}} \right) \tag{3}
\]

4 Experimentation and Results

4.1 Rumourous Event Data

The experimentation is performed on PHEME dataset. PHEME dataset contains a set of collections of Twitter rumored ad non- rumored post which is carried out during breaking news. The breaking news is from a set of news events namely Sydney Siege, German Wings Crash, Charlie Hebdo, and Ottawa Shooting. This dataset is based on rumored news. In PHEME different levels of annotation were done for rumor detection. This dataset contains a Twitter conversation associated with various newsworthy events which include Sydney Siege, German Wings Crash, Charlie Hebdo,
and Ottawa Shooting.

**Table 1** Dataset Details

<table>
<thead>
<tr>
<th>Number of Records</th>
<th>Events</th>
<th>Germanwings-Crash</th>
<th>Charliehebdo</th>
<th>Ferguson</th>
<th>Ottawa shooting</th>
<th>Sydney siege</th>
<th>Consolidated Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 0 (NonRumour)</td>
<td>2494</td>
<td>30923</td>
<td>17696</td>
<td>6436</td>
<td>15320</td>
<td>72869</td>
<td></td>
</tr>
<tr>
<td>Class 1 (Rumour)</td>
<td>1995</td>
<td>7345</td>
<td>6479</td>
<td>5848</td>
<td>8676</td>
<td>30343</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4489</td>
<td>38268</td>
<td>24175</td>
<td>12284</td>
<td>23996</td>
<td>103212</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1** % of records in each class

<table>
<thead>
<tr>
<th>Number of Records</th>
<th>Events</th>
<th>Germanwings-Crash</th>
<th>Charliehebdo</th>
<th>Ferguson</th>
<th>Ottawa shooting</th>
<th>Sydney siege</th>
<th>Consolidated Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 0 (Non-Rumour)</td>
<td>55.56</td>
<td>80.81</td>
<td>73.20</td>
<td>52.39</td>
<td>63.84</td>
<td>70.60</td>
<td></td>
</tr>
<tr>
<td>Class 1 (Rumour)</td>
<td>44.44</td>
<td>19.19</td>
<td>26.80</td>
<td>47.61</td>
<td>36.16</td>
<td>29.40</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 represents the PHEME dataset with each event listed in it, and labels are given for each tasks of rumor detection. Each event varies vastly in size and has a different class label for it.

Table 2 represents the PHEME dataset with percentage of class distribution. Overall, the PHEME dataset contains fewer rumors than non-rumors.

In this research 2 types of the experiment was carried with different subsets of PHEME data set
1. By utilizing 5 events separately
2. By utilizing consolidated data of 5 events

In the mentioned both situation the experiments were performed by 80:20 split of training and testing data.
4.1.1 Ferguson Unrest

The United States people protested for shooting an 18-year-old on August 9, 2014, was an African American named Micheal Brown by US police officers.

4.1.2 Ottawa Shooting

This incident happened in Ottawa’s Parliament on October 22, 2014, resulting in the death of a Canadian Soldier.

4.1.3 Sydney Siege

A gunman captivated almost 10 customers and 8 workers in a candy buffet which was placed in park Road, London on November 9, 2013.

4.1.4 Charles Abraham Hitting

2 sisters were compelled by the police officers of the Deccan chronicle in New York murdered 32 persons and injured 16 persons, on December 4, 2012.

4.1.5 Indonesia Aviation Accident

On March 24, 2015, aviation were picking person from Borobudur to Borneo crashed by killing the crew members and passengers.

4.2 Performance Metrics

4.2.1 Accuracy

It is the percentage of right classifications that a classification algorithms has made when compared to the actual class label in the testing data. Accuracy can be calculated as below:

\[ \text{Accuracy} = \frac{(\text{TNR} + \text{TR})}{(\text{TNR}+\text{TR}+\text{FNR}+\text{FR})} \]

Where, TR is true rumor, TNR is true non-rumor, FR is false rumor, FN is false non-rumor.

If the class label of a record in a dataset is rumor, and the classifier classifies the class label for that record as rumor, then it is called as true rumor. If the class label of a record in a dataset is non-rumor, and the classifier classifies the class label for that record as non-rumor, then it is called as true non-rumor. If the class label of a record in a dataset is rumor, but the classifier classifies the class label for that record as non-rumor, then it is called as false non-rumor. If the class label of a record in a dataset is non-rumor, and the classifier classifies the class label for that record as rumor, then it is called as false rumor.
rumor, but the classifier classifies the class label for that record as rumor, then it is called as false rumor.

Precision=\frac{TR}{TR+FR} \quad \text{Recall}=\frac{TR}{TR+FNR} \quad \text{F1-Score}= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}

5 Results and Discussions

5.1 German wings-Imbalanced Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>62.12%</td>
<td>58.62%</td>
<td>56.67%</td>
<td>57.63%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>72.35%</td>
<td>53.72%</td>
<td>79.27%</td>
<td>64.04%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>73.11%</td>
<td>53.92%</td>
<td>81.25%</td>
<td>64.68%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>73.86%</td>
<td>64.80%</td>
<td>72.81%</td>
<td>70.64%</td>
</tr>
</tbody>
</table>

Table 3 shows the performance of classification algorithms on the dataset based on accuracy, precision, recall and F1-score. The above results show that IAPWU and Pruned Stacking outperform the other two algorithms in most of the parameters. Recall of PSO_AdaBoost is better than Recall of Pruned Stacking.

5.2 German Wings- Balanced Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>70.83%</td>
<td>73.53%</td>
<td>67.57%</td>
<td>70.42%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>72.73%</td>
<td>71.13%</td>
<td>73.19%</td>
<td>72.14%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>73.43%</td>
<td>70.42%</td>
<td>74.63%</td>
<td>72.46%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>74.78%</td>
<td>76.39%</td>
<td>72.85%</td>
<td>74.58%</td>
</tr>
</tbody>
</table>

Table 4 shows the performance of classification algorithms on the dataset based on accuracy, precision, recall and F1-score. Accuracy of AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking in class balanced scenario is good compared to class imbalanced scenario.

Figure 3 represents accuracy and F1-Score of Germanwings_Crash dataset with respect to AdaBoost, PSO_AdaBoost, Improved Adaboost using
Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. Results indicate that imbalanced nature of the data affects the performance of the algorithms.

![Figure 3 Results – German Wings Crash Data](image)

### 5.3 Charliehebdo – Imbalanced Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>83.94%</td>
<td>65.85%</td>
<td>50.94%</td>
<td>57.45%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>88.83%</td>
<td>63.73%</td>
<td>75.00%</td>
<td>68.91%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>90.24%</td>
<td>64.25%</td>
<td>81.58%</td>
<td>71.88%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>90.85%</td>
<td>66.32%</td>
<td>83.12%</td>
<td>73.78%</td>
</tr>
</tbody>
</table>

Table 5 shows the performance of above-mentioned algorithms on the based on accuracy, precision, recall and F1-score. The above results show that IAPWU and Pruned Stacking outperform the other two algorithms in all the parameters.
5.4 Charliehebdo – Balanced Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>85.21%</td>
<td>87.69%</td>
<td>83.01%</td>
<td>85.29%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>91.72%</td>
<td>91.15%</td>
<td>92.08%</td>
<td>91.61%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>92.85%</td>
<td>92.16%</td>
<td>93.34%</td>
<td>92.75%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>93.29%</td>
<td>92.41%</td>
<td>93.96%</td>
<td>93.18%</td>
</tr>
</tbody>
</table>

Table 6 shows the performance of above-mentioned algorithms on the based on accuracy, precision, recall and F1-score. F1-Score of AdaBoost, PSO_AdaBoost, and Improved Adaboost using Probabilistic Weight Updation (IAPWU), and Pruned Stacking in class balanced scenario is great when compared to class imbalanced scenario.

Figure 4 represents accuracy and F1-Score of Charliehebdo dataset with respect to AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. From the results, it is clear that it is necessary to apply class balancing techniques to get the best performance of the classification algorithms.
5.5 Ferguson – Imbalanced

Table 2 Ferguson – Imbalanced - Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>81.65%</td>
<td>72.13%</td>
<td>56.41%</td>
<td>63.31%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>88.04%</td>
<td>62.99%</td>
<td>85.92%</td>
<td>72.69%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>90.56%</td>
<td>68.33%</td>
<td>92.31%</td>
<td>78.53%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>91.10%</td>
<td>70.11%</td>
<td>92.92%</td>
<td>79.92%</td>
</tr>
</tbody>
</table>

Table 7 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. The above results show that Pruned Stacking outperforms the other 3 algorithms in most of the parameters.

5.6 Ferguson – Balanced

Table 8 Ferguson-Balanced - Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>81.11%</td>
<td>83.74%</td>
<td>79.07%</td>
<td>81.34%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>86.96%</td>
<td>80.64%</td>
<td>92.03%</td>
<td>85.96%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>87.75%</td>
<td>82.72%</td>
<td>91.71%</td>
<td>86.98%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>88.78%</td>
<td>83.21%</td>
<td>93.40%</td>
<td>88.01%</td>
</tr>
</tbody>
</table>

Figure 5 Results - Ferguson Data
Table 8 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Accuracy of AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking in class imbalanced scenario is good compared to class balanced scenario. This happens because of characteristics of Ferguson data.

Figure 5 represents accuracy and F1-Score of Ferguson dataset with respect to AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. Results indicate that balanced nature of the data improves the performance of the algorithms.

5.7 Ottawa Shooting – Imbalanced

Table 9 Ottawa shooting – Imbalanced- Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>78.46%</td>
<td>81.54%</td>
<td>76.81%</td>
<td>79.10%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>85.08%</td>
<td>80.07%</td>
<td>91.46%</td>
<td>84.63%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>85.66%</td>
<td>80.43%</td>
<td>92.24%</td>
<td>85.93%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>86.05%</td>
<td>81.14%</td>
<td>92.31%</td>
<td>86.36%</td>
</tr>
</tbody>
</table>

Table 9 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Pruned Stacking and IAPWU gives better results than other 2 algorithms in most of the occasions.

5.8 Ottawa Shooting – Balanced

Table 10 Ottawa shooting – Imbalanced- Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>81.43%</td>
<td>82.25%</td>
<td>75.36%</td>
<td>80.00%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>86.25%</td>
<td>78.65%</td>
<td>92.86%</td>
<td>85.16%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>86.96%</td>
<td>79.00%</td>
<td>94.07%</td>
<td>85.88%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>87.14%</td>
<td>79.36%</td>
<td>94.09%</td>
<td>86.10%</td>
</tr>
</tbody>
</table>

Table 10 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Pruned Stacking and IAPWU gives better results than other 2 algorithms in most of the occasions.
Figure 6 represents accuracy and F1-Score of Ottawa Shooting dataset with respect to AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. Results of class balanced data is better than class imbalanced data.

5.9 Sydney Siege – Imbalanced

Table 11 Sydney Siege – Imbalanced - Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>75.90%</td>
<td>65.57%</td>
<td>67.80%</td>
<td>66.67%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>76.66%</td>
<td>61.66%</td>
<td>72.90%</td>
<td>66.81%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>77.26%</td>
<td>62.45%</td>
<td>73.83%</td>
<td>67.67%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>78.46%</td>
<td>64.43%</td>
<td>75.46%</td>
<td>69.51%</td>
</tr>
</tbody>
</table>

Table 11 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. F1-score of AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking in class balanced scenario is great compared to class balanced scenario.

Table 12 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Pruned Stacking and IAPWU outperform than other 2 algorithms.
Table 12 Sydney Siege – Balanced – Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>76.92%</td>
<td>76.92%</td>
<td>76.92%</td>
<td>76.92%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>82.97%</td>
<td>82.13%</td>
<td>83.54%</td>
<td>82.83%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>85.39%</td>
<td>83.33%</td>
<td>86.90%</td>
<td>85.08%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>86.71%</td>
<td>85.02%</td>
<td>88.00%</td>
<td>86.49%</td>
</tr>
</tbody>
</table>

Figure 7 Results - Sydney Siege Data

Figure 7 represents accuracy and F1-Score of Sydney Siege dataset with respect to AdaBoost, PSO_AdaBoost, Improved AdaBoost using Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. Pruned Stacking and IAPWU algorithms gives better results than all other existing algorithms.

5.10 Consolidated PHEME – Imbalanced

Table 13 PHEME – Imbalanced- Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>82.77%</td>
<td>81.77%</td>
<td>58.87%</td>
<td>68.45%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>82.48%</td>
<td>63.20%</td>
<td>77.87%</td>
<td>69.78%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>83.50%</td>
<td>65.49%</td>
<td>79.32%</td>
<td>71.75%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>85.54%</td>
<td>68.25%</td>
<td>82.56%</td>
<td>73.89%</td>
</tr>
</tbody>
</table>
Table 13 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Pruned Stacking and IAPWU gave better than other 2 algorithms.

5.11 Consolidated PHEME – Balanced

Table 14 PHEME – Balanced- Results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>76.92%</td>
<td>77.72%</td>
<td>76.92%</td>
<td>76.98%</td>
</tr>
<tr>
<td>PSO_AdaBoost</td>
<td>82.97%</td>
<td>82.13%</td>
<td>83.54%</td>
<td>82.83%</td>
</tr>
<tr>
<td>IAPWU</td>
<td>84.52%</td>
<td>85.26%</td>
<td>86.14%</td>
<td>84.58%</td>
</tr>
<tr>
<td>Pruned Stacking</td>
<td>86.25%</td>
<td>86.58%</td>
<td>88.54%</td>
<td>86.35%</td>
</tr>
</tbody>
</table>

Table 14 shows the performance of above-mentioned algorithms on the dataset based on accuracy, precision, recall and F1-score. Accuracy of AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking in class balanced scenario and class balanced scenario is slightly equal. F1-Score of AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking in class balanced scenario is greater than class balanced scenario.

Figure 8 Results – PHEME Consolidated Data
Figure 8 represents accuracy and F1-Score of PHEME Consolidated dataset with respect to AdaBoost, PSO_AdaBoost, Improved Adaboost using Probabilistic Weight Updation (IAPWU), Pruned Stacking algorithms in both class balanced scenario and class imbalanced scenario. Results indicate the importance of handling class imbalance for the purpose of getting good results in terms of accuracy and F1-Score.

6 Conclusions

The rumor propagation model is anticipated through the utilization of SSCA and the results are evaluated based on the similarity between rumor and normal tweets. PHEME the group of information was depending on processing that is held after the designs of the programs were taken out. Various classification algorithms are applied. Various offers are used to differentiate the methodology to analyze the group are finding the gossip, forest which is random, KNN and bayes of native, the labels as gossip and facts. Finally, the Enhanced Adaboost with Particle Swarm Optimization, Improved Adaboost using Probabilistic Weight updation, and Pruned Stacking are proposed to improve the accuracy of predicting rumors. Using the above results, the Pruned Stacking classifier attained a high accuracy rate of 86.25% for the class balanced dataset and 85.54% for the class imbalanced dataset than the other classifiers.

References


Ensemble based Classification of Dynamic Rumor Detection in Social Networks for Green Communication


Biographies

**R.Amutha**, received Bachelor of Science degree from N.K.R Arts College, Periyar University in 2002 and Master of Science degree from Vivekanandha College of Arts & Science, Periyar University in the year of 2007. Now, she is pursuing Ph.D in Nehru Arts & Science College affiliated to Bharathiar University. She has 10 years of experience in teaching. She has published many research articles in international & national journals and conferences. Her Research area mainly focuses on Rumor minimization in social networking data using Data mining Techniques.

**D.Vimal Kumar**, inward Master of Computer Application at K.S.Rangasamy College of Technology, Periyar University India, in 2002. He pursued M.Phil. Computer Science at Kongu Arts and Science College, Bharathiar University in the Year 2007. He was awarded with Ph.D in Anna University in the year of 2014. He has more than 15 years of experience in teaching. He is one of the approved supervisors of Bharathiar University. Currently, 6 research scholars doing their research under his supervision. He has published more than 40 articles in various journals.