MALICIOUS CONTENT DETECTION IN SOCIAL NETWORKS USING HYBRID MACHINE LEARNING MODEL

[1]N. Deepika

Research Scholar, GAT, VTU,
Bengaluru, Karnataka 560098, India
deepikajvijay@gmail.com

[2]N. Guruprasad

Professor, GAT, VTU,
Bengaluru, Karnataka 560098, India
nguruprasad18@gamil.com

Abstract

Social Networking platforms like Facebook, Twitter, Reddit, Weibo, Instagram and many more are the most popular and very easy to use medium for social connectivity, staying up to date with current events and news, relaxing in spare time, sharing opinions on many occurring events etc., Usage of these platforms have tremendously increased year over year say 9 to 12%. As of 2021 half of the percentage of people are using social media out of entire population [1]. With this much usage, if the entire information available in the Network is real and informative then it is really appreciated, but there is clear evidence that there are high chances of dissemination of malicious information for variety of reasons which creates negative impact on the society. So, detecting this type of content in the Social Network is very important research area. From past many years researchers have come up with different ideas to identify this type of information with Data Mining, Machine Learning, Deep Learning techniques. In this paper we propose a hybrid approach HCSTCM (Hybrid Cluster derived Sentiment based Topic Classification Model) to identify malicious content in the Social Network by deriving clusters, sentiments and topic information of the documents, later on using these features for supervised learning. Main aim of the paper is to identify the most dependent features which effect the malicious content without redundancy and improve the classification accuracy. The proposed method is validated with three social platform data.

***Keywords*: Topic Model, clustering, classification, Sentiment Analysis, Social Network Analysis**

1. Introduction

It is vital to identify malicious content which is widely spread across social networks to avoid any evil activity like spoiling one's reputation, deception, improving trending rate, monitory purpose, selfishness, mistakenly, hate towards a person. If we don't identify and stop the spread of malicious content over the network, it unnecessarily creates lot of chaos, fear, misinterpretation and creates negative impact on many gears and spoil the reputation which is not true. As an example, the post in Facebook in 2019 which was viewed more than 150 million times and shared almost 30M times "Trump's grandfather was a pimp and tax evader; his father a member of the KKK". By the time the news is deleted one's reputation would have been severely damaged which no one can reverse its negative consequence. COVID-19 pandemic time the remedies and medicines which have been spread across the social networks were clueless. Lot of confusion has been created among the citizens including highly graduated people. So, it is very much indispensable to develop a machine learning model which can effectively identify malicious content in the social network. This identification can be done in many ways. so far, many researchers have done lot of research with diverse techniques. The approaches are based on,

1. Using Hybrid approaches DT-SVM, CSI Model, HFCM Model [19,28]
2.Unsupervised learning-based approaches [13,15,19,28]
3.Using ensemble approaches Voting classifier, Boosting classifier, RF [6,28]
4.Using content-based features [29] like count of hashtags, mentions, URLs used, number of likes, text, images, title etc.,

5.Using context-based features [16] like sentiment of the message, linguistic context, semantics, syntax, writing pattern, number of exclamation/punctuations used etc.,
6.Using standalone Machine Learning [1, 24] /Deep Learning models like Naïve Bayes, SVM, clustering, LSTM, BERT, CNN [14]
7.Using user-based features [3] like count of followers, timely tweet count, retweet count, age of account, friends count, geo enabled, verified etc.,

i). In our work we propose a novel based malicious content detection hybrid approach which is amalgamation of content-context-based features, sentiment, and Topic. It has been proven with much research that hybrid model usually outperforms [3] standalone model. In our approach we have done investigation on fake news data. The study reveals that fake news is related to few important features,

ii). Malicious news will have a strong sentiment positively or negatively to influence the readers emotion which will eventually take them forward to read and reply and share the post. So, including the sentiments will certainly improve the model performance [4].

iii). Malicious news would spread in unequal proportion with the topic. Work done by [22] also confirms that the topic derived from the news and contextual information have certain impacts of fake news detection task. So, we understood that extracting topics from the data will certainly improve the performance of the model.

iv). As Hybrid model usually outperforms the standalone model, we have first implemented unsupervised model Bisecting Kmeans clustering with Elbow technique and derived the labels. Later these labels also will be used as one of the features for the final model.

v). Finally, a classification model with Random Forest and SVM have been trained with the set of features: text features, content and context features, Sentiment information, Topics derived for each sentiment and labels derived from Kmeans clustering.

2. Related Work

Identifying malicious content has interested many researchers due to its dynamic problem nature. And it is highly important to address this problem, otherwise it could create severe negative effect. In this section we will briefly acknowledge the prior work done regarding malicious content detection. Qing Liao et al. [22] has developed an integrated model with multitask learning with N-Graph, topic, and con-textual information on LIAR Dataset. They have considered the fact that few topics have high probability of being fake and few authors have high probability of publishing the fake news.

2.1. *Malicious content detection using deep learning models*

Social article fusion model was adopted to classify fake news by Shu et al [26] with a deep learning approach by using metadata such as language based, replies, retweets etc. Bhavika et al. [4] used Sentiments along with text features to identify fake content on Twitter data using Naïve and Random Forest algorithms. Aba Babakura et al. [1] used Deep learning techniques with diverse features (text features, handcrafted features) and achieved 99.3% F1 Score. Balasubramanian et.al. [3] has addressed using features from both text and visuals by proposing CB-Fake model. Using BERT, they have extracted context based textual features from news articles and by using Caps net model, extracted visual features and the fusion of both features are then passed to classification. CB-Fake model has given 93% and 92% accuracy on PolitiFact and Gossip cop datasets respectively. Many researchers have adopted DL models because of automatic feature extraction [3,7,14,16]. Kaliyar et al. [14] proposed Fake BERT model, which is a combination of BERT and Convolutional Neural Network which handles the textual contents in a bidirectional way. Shu et al. [17] proposed Co-attention mechanism to discover top K important sentences and top K important user reviews to classify fake news.

Ma et al. [20] were the first one to introduce Recurrent Neural Network (RNN) model for textual data sequence for detecting rumors over time. Wang [24] introduced a new data set, called LIAR dataset, proposed a hybrid deep learning model in which CNN for the textual-feature extraction and Bi-LSTM for meta-data feature extraction is used. Yin et al. [27] employed principal component analysis (PCA) and CNN for feature extraction from the news contents.

2.2. *Malicious content detection using standalone approaches*

Ahmed et al. [2] introduced a new dataset ISOT which is based on real-world sources, for FND they have apprised n-gram analysis with TF-IDF for feature vector representation. Prateek Dewan et al. [21] contributions from 2014 to 2016 towards detecting malicious posts characterizes malicious content generated on Facebook during news making events. They have experimented with a dataset of 4.4 million public posts collected from Facebook APIs using Random Forest classifier. Lawrence et al. [15] have done research to analyze crime data using 3 different machine learning algorithms Linear regression, Additive linear regression, and Decision stump model. Hailu et al. [9] investigated to detect spam profiles and spam messages respectively on public group posts dataset using traditional machine learning algorithms and obtained accuracy of 99.26% and 99.7% with Random Forest model.

2.3. *Malicious content detection using hybrid approaches*

Natali et al. [19] has developed CSI hybrid deep model which first captures temporal textual, user features to measure the response and text. Later Suspicious score of each user is calculated and finally these two features are integrated, and label is predicted. The proposed hybrid approach is the decision-making model which consists of multiple machine learning algorithms along with crowdsourcing with human touch. Performance is better with hybrid model instead of using only ML algorithms. Rahman et al. [24] proposed hybrid approach for anomalous user detection DT-SVMNB using user content and behavior and achieved accuracy of 97.7%.

3. Data collection

For our research we have collected and created three different datasets from Face-book, Twitter and Reddit using python packages and their corresponding methods. If any message/post has less than four words, then it is considered as general message and removed from the database.

3.1. *Dataset1: Facebook posts*

Unlike other social networks Facebook does not provide an API endpoint to gather an incessant random sample of public posts in real time. Thus, we have created few accounts and using the below procedure FB posts are scrapped. Our first dataset is created from Facebook posts using Facebook scrapper tool with python script. Facebook scrapper is a scrapping tool which is engineered to scrape data from our own Facebook account or public Facebook pages. Extracted data includes post id, post, shared text, timestamp, likes, comments, shares, user id, username, user URL, reviews etc., The procedure used to collect data is explained in two steps. As a first step we have created cookies.txt file with created Facebook account. Second, written python script with "facebook_scraper" package and "get_posts" method by passing the search string along with cookies and pages attributes. Obtained data is saved into csv file. An example screen shot of extracted data with search string as BBC News is shown below. For ensuring protection of data privacy datasets are not shown in the paper.

As this data is not labelled, we have used human annotations by cross checking with several newspaper contents like headlines, visuals, or captions to understand its reliability. If the extracted data is not supporting the content of certain list of newspapers, text is labelled as "mostly false" else it is labelled as "mostly true". Final dataset is prepared by merging the scraped data with BuzzFeed-Webis Fake News Corpus 2016 which contains 1627 Facebook posts from 9 publishers on 7 workdays close to the US 2016 presidential election. All publishers earned Facebook's blue checkmark, indicating authenticity and an elevated status within the network. Each post and linked news article were rated "mostly true", "mixture of true and false", "mostly false", or "no factual content" by BuzzFeed journalists. Final dataset has 9199 records in total.

3.2. *Dataset1: Twitter posts*

The second dataset has been extracted from Twitter platform using pythons "tweepy" package and cursor method. The method empowered for extracting data from twitter is explained here. As a first step we need to get twitter developer account. To access twitter API, we need 4 different tokens-secret keys: consumer key and secret key, Access token and Access token secret key. These can be obtained from twitter developer account. As we are going to extract fields like 'created at', 'name', 'verified', 'text', 'description', 'location', 'followers count', 'friends count', 'statuses count' etc., first we need to apply for twitter elevated access. After approval from twitter team, we can scrape this information. Tweets are queried with a given string and extracted for every seven days on a given string as that is the allowed days limit in twitter using "tweepy. Cursor" method. We have used lang as 'en' to avoid non-English language messages. Dataset has 7891 records in total. The below screenshot is the example data scrapped from Twitter.

As this data is not labelled, we have used human annotations by cross checking with several newspaper contents like headlines, visuals, or captions to understand its reliability. If the extracted data is not supporting the content of any newspaper is labelled "mostly false" else, it is labelled as "mostly true". Final dataset is prepared by combining the scraped data with Kaggle Twitter Fake News dataset.

3.3. *Dataset1: Reddit posts*

The third dataset which we have gathered is from Reddit social network platform. Using python script and a package called “praw” and Reddit method, reddit data is scrapped. After creating an account in reddit we should first obtain “client id”, “client secret” and “user agent” keys from reddit. By using subreddit method and its value as “The onion” and “usa news” we have scraped post title, score, id, subreddit, url, number of comments, date created. “The onion” data is usually sarcasm, so it is labelled as “mostly false” and “usa news” data is legitimate and is labelled as “mostly true”. Total dataset has 29,113 records as total.

4. Methodology

The proposed solution for malicious content detection is a two layered approach. In the first layer we have derived contextual features from Global vector model, unsupervised labels are derived from K means clustering, sentiments are derived from Vader model and topics are derived from Top2vec model. Fusion of all these features is passed to the second layer i.e., classification layer. Final Feature set is represented as below. Where C1…Cn are contextual features of the tweet/post, L1…Lk are cluster to which the tweet/post belongs to, S1…S3 are three sentiments, T1…Tm are the topic to which the tweet/post belongs to.

$$F = \begin{bmatrix} C1 \ldots. & \ldots.Cn & L1 \ldots Lk & S1 \ldots S3 & T1 \ldots Tm \\ C2 \ldots. & \ldots.Cn & L1 \ldots Lk & S1 \ldots S3 & T1 \ldots Tm \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Cm \ldots. & \ldots.Cn & L1 \ldots Lk & S1 \ldots S3 & T1 \ldots Tm \end{bmatrix} \tag{1}$$

Detailed explanation is provided in further sessions. Proposed solution is depicted in Fig. 4.

4.1. *Algorithm*

1. Extract data (DS1) from social Network and clean the data.
2. Do necessary preprocessing and get the labels by comparing with news content.
3. As the size of the data is less after preprocessing, we have merged DS1 with existing datasets DS2.The final dataset is DS3
4. Clean final dataset.
5. Extract,
   a. Textual features like length of the text, question marks, exclamations etc.
   b. Language specific features like slangs, nouns etc.
   c. User specific features like retweets, likes, friends, followers etc.
6. Create subset of features with Fisher score (FS) feature selection analysis by remove redundant and least important features.
   FS of nth feature is calculated as,
   $$FS(n) = \frac{\sum(\mu_{nk}-\mu_n)^2 N_k}{\sum N_k * x^2{}_{nk}} \tag{2}$$
   here, μ_{nk} is mean and $x\ _{nk}$ is variance of nth feature in kth class, N_k is kth class total count and μ_n is nth feature mean.
7. Generate sentence vectors using word2vec word embeddings.
8. First layer of Hybrid model ‘unsupervised model’ is done to get the labels based on the content similarity using Bisecting Kmeans algorithm. add these label as one of the features to DS3.
9. predict the sentiments of each document and add this as next feature to DS3.
10. Use topic modelling to find topic numbers of each document and include this as next feature to DS3.
11. Second layer of hybrid model ‘classification model’ is trained with DS3 to predict the malicious content.

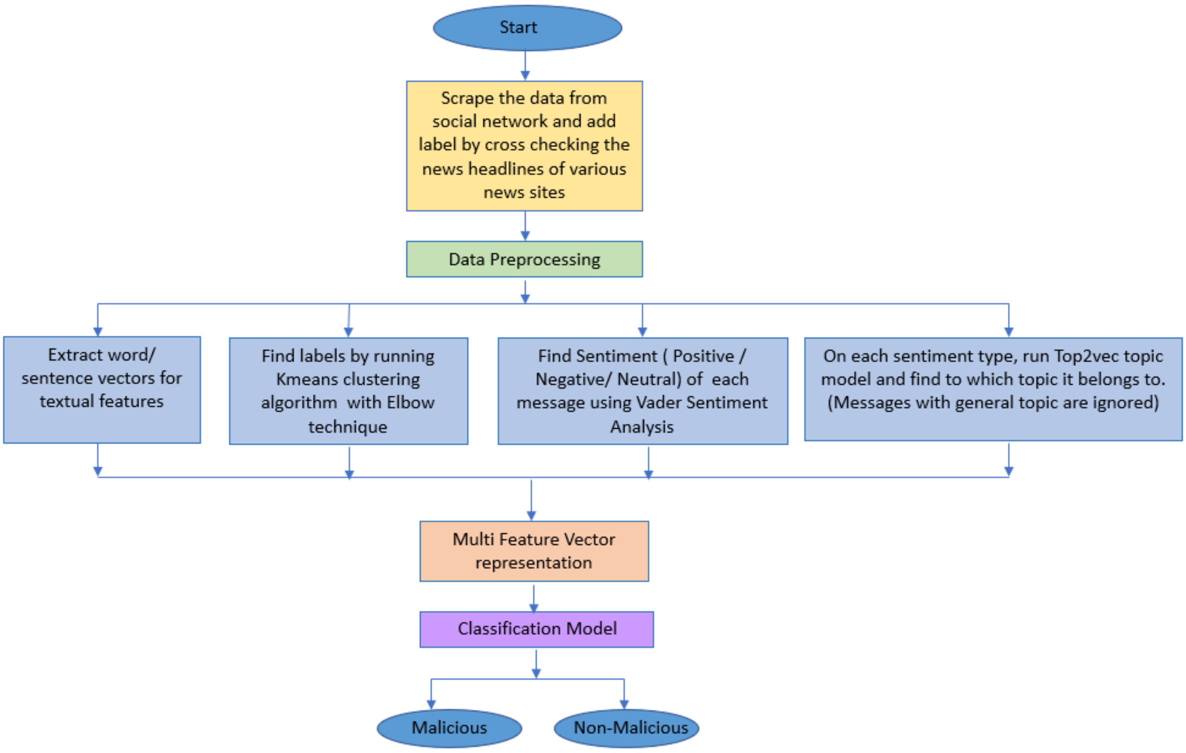
Start
Scrape the data from social network and add label by cross checking the news headlines of various news sites
Data Preprocessing
Extract word/ sentence vectors for textual features
Find labels by running Kmeans clustering algorithm with Elbow technique
Find Sentiment (Positive / Negative/ Neutral) of each message using Vader Sentiment Analysis
On each sentiment type, run Top2vec topic model and find to which topic it belongs to. (Messages with general topic are ignored)
Multi Feature Vector representation
Classification Model
Malicious
Non-Malicious

Fig. 1. Example Proposed solution for malicious content detection using hybrid approach

As an initial step, the prepared data set is preprocessed with tasks like converting data to lower case, removing punctuations, replacing contractions, selected stop words removal, spell corrections with text blob package, converting emojis and emoticons, splitting hashtags, mentions, URLs from posts, tokenization, Lemmatization etc. have been done.

4.2. *Word embeddings*

It is very important to choose suitable vectorizer model for better performance of the ML/DL Model. The word vectors have been obtained from preprocessed text using tfidf vectorization, word2vec model and Glove Model. Glove Model have given better similar words compare to other models.

4.2.1. *tfidf Model*

Term Frequency Inverse Document Frequency (Tfidf), which is offered in sklearn's package is the first algorithm we tried to transform text to vectors. Tfidf is a vectorization technique which compares number of times a word appears in a document with the number of documents a word appears in thus giving the authenticity of a word.

$$TF\text{-}IDF = TF\ (term,\ doc) \times\ IDF\ (term), \tag{3}$$

Where TF (t, d) = $\sum_{w \in d} frequency(w,t)$ and
$frequency(w,t) returns\ 1\ if\ w\ is\ same\ as\ t\ otherwise\ returns\ 0$

$$IDF\ (term) = log\ (\frac{no.of\ docs}{1+doc\_freq(doc,term)}) + 1 \tag{4}$$

Even though similarity between documents is easily calculated, this model has a limitation of not capturing semantics, co-occurrences, and position in text. e.g., orange and apple are two separate words for Tfidf.

4.3. *Word2Vec model*

Another embedding model tried to obtain word vectors in the research is learning based Word2vec Model which is found in Genism's library. Word2vec uses three-layer shallow neural network to create lower dimensional word vectors. The two flavors of word2vec are skip gram (given target word, Context is found) and continuous bag of words (given context, target word is found). Words with similar context will be placed close in the vector space and vice versa.

$$P\,(W_i\,|W_j) = \frac{\exp\,(VWj.VWi^T)}{\sum_{k=1}^{V}\exp\,(VWj.VWk^T)} \tag{5}$$

Though it captures semantics, context, and co-occurrences in the text, but the view of the data is limited to local, and sub linear relationships are not explicitly defined. For our dataset word2vec model has given good similarity.

4.4. *Unsupervised learning*

In the process of obtaining features for classification model, first we have obtained labels for the dataset based on the similarity using unsupervised approach.

4.4.1. Bisecting K Means Clustering

BKC is a combination of well-known Kmeans and divisive hierarchical clustering [25]. Applying Bisecting KMeans clustering as an initial layer which is to identify the similarities among attributes and derive clusters such that, within the cluster similarity is much higher than between clusters. As each document is represented in high dimension space, optimal number of clusters are identified using silhouette method [25]. For each K value obtained train the BKC model and find silhouette score. With all the obtained values plot the graph with y axis as silhouette value and number of clusters(k) on x-axis. Wherever we have steep and sharp fall in the plot that is the best value to run the model.

Bisecting KMeans algorithm

1. Initialize k value as sum of squared error vs clusters graph, and I number of iterations.
2. First bisection is done by splitting the entire data into two clusters C1 and C2 with random centroids using Kmeans approach. Data points are assigned to the cluster whose mean has least squared Euclidean distance.

$$\min_{c_i \in C}(\,\mathrm{ED}(c_i,x)^2) \tag{7}$$

where, ED is Euclidean distance

```
3. for i in range(k):
        ignore cluster with less noise
        bisect most noisy cluster and form two clusters C1 and C2
```

Number of clusters are analysed with Silhouette analysis to find out how close each data point in first cluster to the data points in second cluster. The more the Silhouette score, the best the cluster formed.

$$sil(i) = \frac{b(i)-a(i)}{\max\,(a(i),b(i))} \tag{8}$$

here,
a(i) is mean intra cluster distance to all data points which confirms how well a datapoint is assigned to its own cluster.
b(i) is mean inter cluster distance from a datapoint in cluster Ci to Cj, i.e., average dissimilarity to the nearest cluster.

4.5. *Sentiment analysis*

After deriving the first feature with unsupervised clustering, next feature of the hybrid model is finding the sentiments of the documents. Comparatively very few researchers considered sentiment analysis in finding the malicious content. But few research papers [4, 10, 18] confirm that sentiment either as main feature or as partial feature, plays a major role in fake content detection. To spread fake content many users, exaggerate real content by tweaking it as either too much positive or too much negative or create fake negative or positive content for various purposes. This is the main clue for utilizing sentiment in fake content identification.

4.5.1. Vader sentiment analyzer

Valence aware dictionary and sentiment reasoner [18] is especially optimized for social media data based on lexical and rule-based features. We have done minimum text cleaning as incorporating heuristics like, 1. Degree modifiers of the word such as extreme, very, too etc., 2. Slang of the text yeah, nah etc., 3. Western style emoticons :), :(, etc., 4. Acronyms like LOL, SD, etc., 5. Upper case text etc. 6. Contrastive conjunctions like but, so, etc., will greatly help in increasing the magnitude of intensity of the sentiment thus improving model performance.

Few words especially negated words are not treated properly with Vader, we have created a dictionary of words with the sentiment score. Vader calculates valance score using heuristics. These valance scores are summed up and normalized between -1(most negative) and +1(most positive) to get compound score of documents.

$$x = \frac{x}{\sqrt{x^2 + \alpha}} \tag{9}$$

here x is sum of valance scores of constituent words and α is normalization constant with 15 as default value. Once we got the compound scores, sentiments are calculated. Python code snippet is shown below.

```
def sentiment_cal (row, thresh=0):
    if row. vaderscore < thresh:
        return “negative”
    elif row. vaderscore > thresh:
        return “positive”
    else:
        return "neutral"
```

4.6 *Topic generation*

Malicious content is not same across different topics. Since we have already derived the sentiments of each document, now we aimed to identify the topics in positive, negative, and neutral sentimental documents to get more hidden patterns in the documents using topic model. We have observed that malicious content distribution varies from topic to topic, so if we include topic information as one feature then final model performance can be further increased. Topic Modelling is unsupervised ML technique. Topics can be assumed as discrete values obtained by using latent semantic structures e.g., politics, disease, movies, science, education, etc. LDA (Latent Dirichlet Allocation) and Top2Vec are two most popular topic modelling techniques to identify topics in the given corpus.

Top2vec assumes that all the documents which have similar topic certainly creates dense area in the jointly embedded document and word semantic space. The topic vectors are calculated as the centroids of each dense area of document vectors. A dense area is an area of very likely documents and average document mostly representing that area is topic vector. As each document could belong to more than one topic, based on the maximum topic weight and document partitioning each document d ∈ D is assigned to exactly one topic t ∈ T [5]. We have extracted more information and representation of the documents with Top2vec model.

4.6.1. Top2Vec

The functionality of Top2vec model is through, 1. Generating embedding vectors for documents and words such that similar documents are brought close 2. Reducing dimensions using UMAP 3. Find dense areas of documents using HDBSCAN clustering 4. Extract topic vectors from each cluster centroid. After assigning each document to exactly one topic, these topics are evaluated using probability weighted amount of information gain *PWI.*

$$PWI(T) = \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d,w)\, log(\frac{p(d,w)}{p(d)p(w)}) \tag{10}$$

Here ‘t’ is single topic from list of topics T, d is single document from list of documents D in topic t, w is word from list of words W of topic t, $p(w)$ is marginal probability of w across D, pointwise mutual information between w and d is calculated from long term.

4.7 *Classification*

The second layer of the approach is training a classification model for identifying content into malicious and non-malicious with all the derived features. The final model corresponds to a function that takes input as fusion of text, cluster, sentiment, and topic features of a document $d \in D_f$ and produces output $d \in C_{[m,n]}$, where C is the classification which is either m malicious or n non-malicious. We have experimented with LSTM neural network with 5 epochs, 20% dropout rate and sigmoid activation function with Adam optimizer. Another two classification models which we have experimented are ensemble bagging Random Forest classifier with default parameters and Support Vector Machine by tuning kernel coefficient and regularization parameter.

5. Experimental Results

The final dataset is fabricated as explained in section 3 with 30843 records in total before cleaning and after removing general messages and short messages we have 10000 records. Equal portions of malicious and non-malicious posts were made. Frequency of the terms in both categories of the dataset were visualized using word cloud representations. Fig 5 depicts malicious and non-malicious content word clouds.

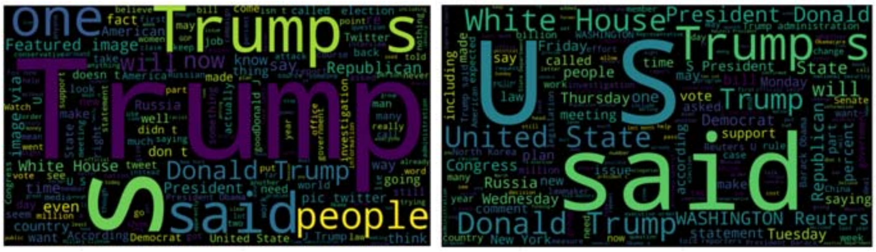

Fig. 2. Non-malicious and Malicious posts – word cloud

After preprocessing the data, the first layer of the approach clustering is performed using Bisecting Kmeans algorithm to derive the class labels. Fig 6 depicts the derived classes. The number of clusters are decided based on Elbow technique.

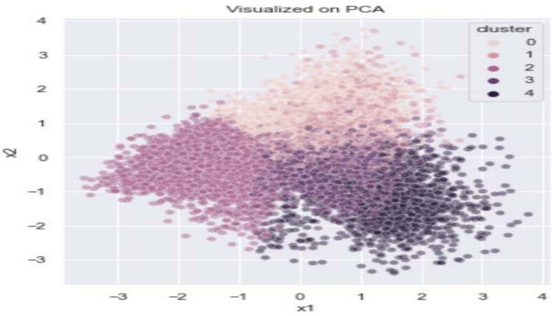

Fig. 3. Class labels derived from BKM clustering

After deriving the cluster labels, sentiment analysis is performed to get the polarity and integrity of the posts thus finding the sentiments of each document. Count of positive, negative, and neutral sentiments in each class is identified. As expected, in malicious content there are both positive and negative posts and no neutral posts but in non-malicious posts there are more positive posts, very fewer negative posts and few neutral posts. This is one of the reasons of including sentiment as one of the substantial features to the final model. Sentiment scores of all the three sentiment types is depicted in Fig 7.

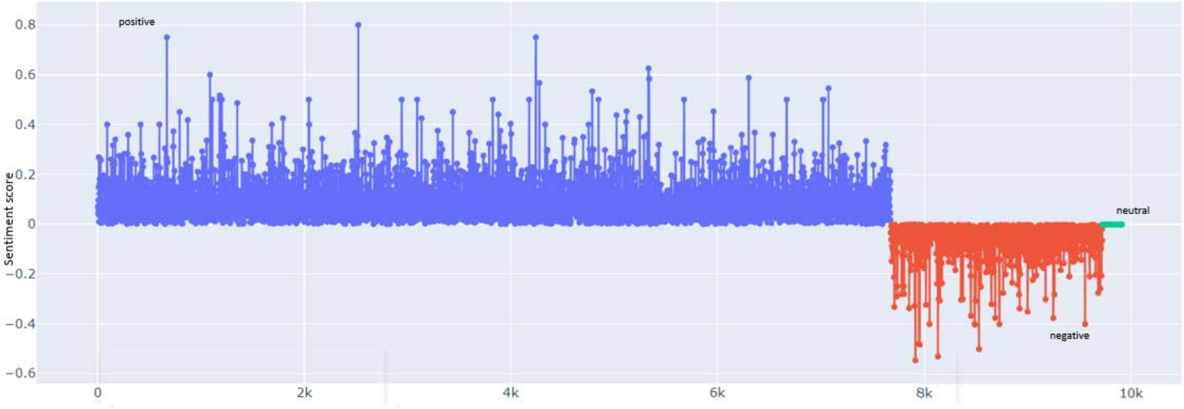

Fig. 4. Classes derived from BKM clustering

As a final step in layer1 Topic modelling is performed to generate the topic numbers from each class for malicious and non - malicious posts from all the sentiments. With this the fusion of features are readily available for layer1. All these features are included in the second layer of the approach to classify the content. Generated topics for non-malicious documents with positive and negative sentiments is depicted in figure 8 which is shown below.

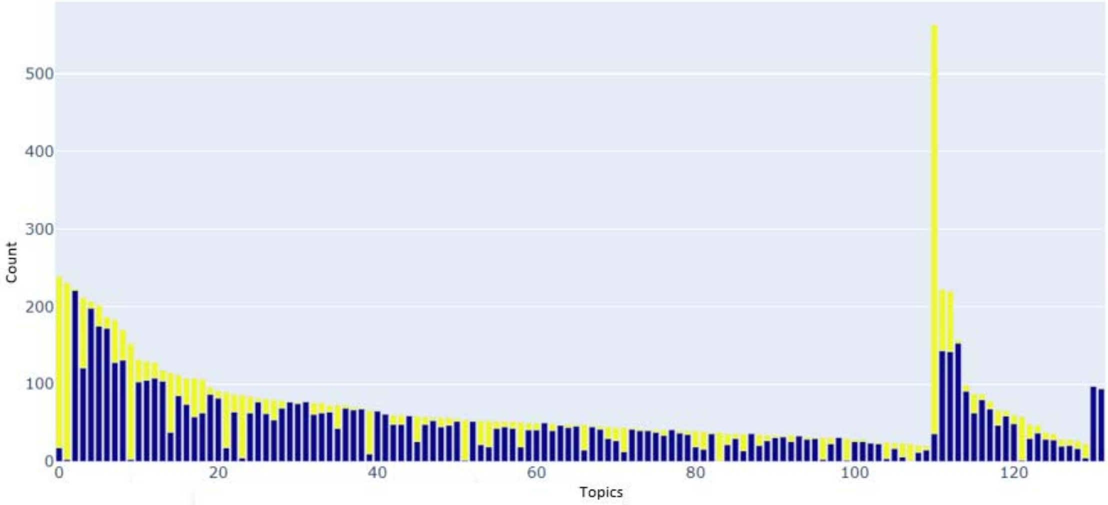

Fig. 5. Topics derived on non – malicious posts for positive and negative sentiments

In second layer classification model is applied with derived features. In the research we have experimented two supervised learning models random Forest and Support vector machine with only text features, combination of text features and cluster labels, combination of text features, cluster labels and sentiments and finally combination of text features, cluster labels, sentiments and topics obtained. Among all the last approach proved to be performing better with respect to accuracy. 70% of the dataset was used to train the data and 30% of data is used for testing. To evaluate the model performance, we have used different metrics mostly based on confusion matrix. Confusion matrix on test set consists of mainly four measures namely True positives: malicious posts which are identified as malicious, False positives: non-malicious posts identified as malicious, True negatives: non-malicious posts identified as non-malicious, False positives: non-malicious posts identified as malicious. The results of dataset1 are shown in Table1.

Data Set	Features used	Algorithm	Class	Accuracy%	precision%	Recall%	F1-Score%
Facebook Posts	Text features	RF	Non-malicious	0.95	0.94	0.99	0.99
			Malicious		0.98	0.86	0.86
		SVM	Non-malicious	0.95	0.94	0.99	0.99
			Malicious		0.98	0.86	0.86
	Text Features + BKM	RF	Non-malicious	0.96	0.96	0.98	0.97
			Malicious		0.96	0.91	0.94
		SVM	Non-malicious	0.97	0.98	0.99	0.98
			Malicious		0.97	0.95	0.96
	Text Features + BKM + Sentiments	RF	Non-malicious	0.968	0.97	0.98	0.98
			Malicious		0.96	0.94	0.95
		SVM	Non-malicious	0.987	0.99	0.99	0.99
			Malicious		0.99	0.97	0.98
	Text Features + BKM + Sentiment + Topics	RF	Non-malicious	0.972	0.98	0.98	0.98
			Malicious		0.97	0.95	0.96
		SVM	Non-malicious	**0.99**	0.99	1.00	0.99
			Malicious		0.99	0.97	0.98

Table 1. Comparison of results with various features on dataset1

The experiment is performed with dataset1 and dataset2, and in both the cases SVM outperformed Random Forest with fusion of text, cluster labels, sentiment, and topic features. The experimental results of dataset2 are shown in Table2.

Data Set	Features used	Algorithm	Class	Accuracy%	precision%	Recall%	F1-Score%
Twitter tweets	Text features	RF	Non-malicious	0.70	0.71	0.84	0.77
			Malicious		0.68	0.49	0.57
		SVM	Non-malicious	0.72	0.72	0.86	0.78
			Malicious		0.74	0.65	0.73
	Text Features + BKM	RF	Non-malicious	0.77	0.75	0.90	0.82
			Malicious		0.82	0.62	0.70
		SVM	Non-malicious	0.78	0.76	0.88	0.83
			Malicious		0.81	0.67	0.73
	Text Features + BKM + Sentiments	RF	Non-malicious	0.78	0.77	0.89	0.83
			Malicious		0.81	0.66	0.73
		SVM	Non-malicious	0.79	0.79	0.89	0.85
			Malicious		0.82	0.69	0.75
	Text Features + BKM + Sentiment + Topics	RF	Non-malicious	0.816	0.84	0.85	0.85
			Malicious		0.77	0.76	0.77
		SVM	Non-malicious	**0.82**	0.85	0.87	0.88
			Malicious		0.87	0.78	0.79

Table 2. Comparison of results with various features on dataset2

Accuracy: How much percentage of posts are accurately predicted as malicious and non-malicious.

$$Accuracy = (TP+FP)/(TP+TN+FP+FN) \tag{11}$$

Recall: Number of posts predicted as malicious out of all malicious posts.

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

Precision: Out of all predicted malicious posts, how many are malicious is identified with precision score.

$$precision = \frac{TP}{TP+FP} \tag{13}$$

F1-Score: The tradeoff between precision and recall is identified with F1-score. It considers both False positives and False negatives to calculate the harmonic mean.

$$F1-Score = 2\times\frac{Precision\times Recall}{Precision+Recall} \tag{14}$$

6. Conclusions and Future Work

Past many years researchers have been working to detect malicious content in social networking sites with various approaches. In this paper we have presented our contribution for finding malicious posts with hybrid approach HCSTCM (Hybrid Cluster derived Sentiment based Topic Classification Model). Our hybrid approach identifies most dependent fusion of features using clustering, sentiment analysis and topic model which effects the malicious content without redundancy and these futures are used in the classification to improve the accuracy. The proposed method is validated with three social networking platform datasets and the results were discussed in experiment section. Among all datasets Facebook dataset provides highest accuracy and Reddit with lowest accuracy. Among all classification models SVM with Tfidf vectors, Bisecting Kmeans Clusters, Vader sentiments and Top2vec topics has acquired best results. As part of feature work, we are going to experiment with deep learning models to improve accuracy.

Funding

No funding is provided for the preparation of manuscript.

Conflict Of Interest Statement

The authors have no conflicts of interest to declare.

References

[1] Aba Babakura et al.Improved Method of Classification Algorithms for Crime Prediction, IEEE International Symposium on Biometrics and Security Technologies PP,250-255, (2014)

[2] Ahmed H, Traore I, Saad S (2017) Detection of online fake news using n-gram analysis and machine learning techniques. International conference on intelligent, secure, and dependable systems in distributed and cloud environments. Springer, Cham, pp 127–138

[3] Balasubramanian Palani, Sivasankar Elango, Vignesh Viswanathan K, CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT, Springer, Multimedia Tools and Applications, 10.1007/s11042-021-11782-3 (2021)

[4] Bhavika Bhutani,et al, Fake News Detection Using Sentiment Analysis, 9 78-1-7281-3591-5/19/ IEEE (2019).

[5] Dimo Angelov, Top2vec: Distributed Representations of Topics, arXiv:2008.09470v1 [cs.CL] 19 Aug (2020)

[6] Domenica Fioredistella Iezzi, Mario Mastrangelo,The IEMA Fuzzy c-Means Algorithm for Text Clustering, JADT : 12es Journées internationales d'Analyse statistique des Données Textuelles (2014)

[7] Dun li, Haimei guo, Zhenfei wang, and Zhiyun zheng, Unsupervised Fake News Detection Based on Autoencoder, IEEE Access, Volume9, pp 29356 – 29365 (2021)

[8] Guo C, Cao J, Zhang X, Shu K, Liu H (201: Learning dual emotion for fake news detection on social media (arXiv preprint). arXiv: 1903. 01728

[9] Hailu Xu, Weiqing Sun, and Ahmad Javaid. Efficient spam detection across online social networks., Big Data Analysis (ICBDA), IEEE International Conference on, pages 16. IEEE, (2016)

[10] Hutto C.J, Gilbert E.E, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of social media Text. Eighth International Conference on Weblogs and social media (ICWSM-14). Ann Arbor, MI, June (2014)

[11] Iftikhar Ahmad,et al, Fake News Detection Using Machine Learning Ensemble Methods, Hindawi Complexity, volume 2020, Article id 8885861, 11 pages, DOI 10.1155/2020/8885861 (2020).

[12] Jeffrey Pennington, Richard Socher, Christopher D. Manning, GloVe: Global Vectors for Word Representation (2014)

[13] Jin Xiao, et al, A Hybrid Classification Framework Based on Clustering, IEEE Transactions On Industrial Informatics, VOL. 16, NO. 4, APRIL (2020)

[14] Kaliyar RK, Goswami A, Narang P (2021) FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools and Applications 80(8):11765–11788 (2021)

[15] Lawrence McClendon and Natarajan Meghanathan, using machine learning algorithms to Analyze crime data,Machine Learning and Applications: An International Journal (MLAIJ) Vol.2, No.1, March ,PP.1-12 (2015)

[16] Ma J, et al, Detecting rumors from micro blogs with recurrent neural networks (2016)

[17] Md. Shafiur Rahman, et al, An efficient hybrid system for anomaly detection in social networks, Springer open Cyber Security, https://doi.org/10.1186/s42400-021-00074-w (2021)

[18] Miguel A. Alonso,et al, Sentiment Analysis for Fake News Detection, Electronics 2021, 10, 1348. https://doi.org/10.3390/electronics10111348 (2021)

[19] Natali Ruchansky, Sungyong Seo, Yan Liu, CSI: A Hybrid Deep Model for Fake News Detection, Session 4B: News and Credibility, ACM, DOI 10.1145/3132847. 3132877 (2017)

[20] Potthast, M, et al, A Stylometric Inquiry into Hyperpartisan and Fake News. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; Volume 1: Long Papers; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 231–240. [CrossRef]

[21] Prateek Dewan, Ponnurangam Kumaraguru, Facebook Inspector (FbI): Towards Automatic Real Time Detection of Malicious Content on Facebook, Social Network Analysis and Mining volume 7, Article number: 15 (2017)

[22] Qing Liao, et al, An Integrated Multi-Task Model for Fake News Detection, DOI 10.1109/TKDE.2021.3054993, IEEE Transactions on Knowledge and Data Engineering (2021).

[23] Sharma K, et al, Combating Fake News: A Survey on Identification and Mitigation Techniques. ACM Trans. Intell. Syst. Technol. (2019)

[24] Shu K, et al, Defend: Explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 95–405 (2019)

[25] Silhouette Method — Better than Elbow Method to find Optimal Clusters | by Satyam Kumar | Towards Data Science (2020)

[26] Wu L, Liu H Tracing fake-news footprints: Characterizing social media messages by how they propagate. In: Proceedings of the eleventh ACM international conference on web search and data mining, pp 637–645 (2018)

[27] Yin L, Meng X, Li J, Sun, Relation extraction for massive news texts. Comput Mater Continua 58:275–285 J (2019)

[28] Z. Muda, et al, K-Means Clustering and Naive Bayes Classification for Intrusion Detection, Journal of IT in Asia, Vol 4 (2014)

[29] Zhou X, et al, Fake news: Fundamental theories, detection strategies and challenges. In: Proceedings of the twelfth ACM international conference on web search and data mining, pp 836-837 (2019)

Authors Profile

N.Deepika is currently pursuing PhD at VTU. She is a post graduate fron JNTU, Hyderabad.She has more than 20 years of academic and industry experience with more than 20 publications. Her resarch interests include Machine Leraning, Data structures, Algorithms, Natural language processing, Cloud architectures, etc. Email: deepikajvijay@gmail.com

Dr.N.Guruprasad is corrently working as a proffessor at GAT, Bangalore. He has more than 25 years of experience with more than 60 publications and 6 books in CSE. He is a senior member of various societies and profound speaker at various Engineering colleges and industries.His areas of interests include, Operations Reserch, Data Mining, Machine Learning, Computer networks, Algorithms and Data Structures etc. Email: nguruprasad18@gmail.com