IJO -INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND ENGINEERING

(ISSN: 2814-1881) Dr Pankaj Agarkar *

https://ijojournals.com/ Volume 07 || Issue 11 || November, 2024 ||

Ensuring Social Media Authenticity: Leveraging Machine Learning

Dr Pankaj Agarkar, IIP PDF Innovator Eudoxia University USA,

and HOD Computer ,ADYPSOE Lohegaon, Pune, India

Prof. Nita Kale , Comp dept ADYPSOE Lohegaon , Pune, India

Dr Anita Mahajan , PG Head computer , ADYPSOE , Pune, India

Dr Niraja Jain , Director IQAC , MIT ADT , Pune, India

Dr R G Konnur , Director Eudoxia University ,USA

Abstract: Identity fraud continues to be a major concern in contemporary online social networks. Research initiatives emphasize developing technologies to detect identity fraud; however, their effectiveness often depends on empirical validation. This study explores identity fraud detection through clustering and classification methods, aiming to overcome traditional methodological shortcomings and propose enhancements for practical application. Data is collected from social media accounts and processed through steps such as Natural Language Processing (NLP), vectorization, dimensionality reduction, and data normalization. Behavioural analysis and profile attributes are leveraged for feature extraction. Profiles are classified as genuine or fraudulent using clustering approaches, followed by deep learning classification on similar datasets.

Keywords: Social media, fake news, twitter accounts, natural language processing, clustering, grouping, RNN(Recurrent Neural Network)

Introduction

Public engagement platforms such as Twitter and Facebook are vital for digital communication and collaboration. The spread of false information through automated bots presents a substantial challenge on these platforms. Research suggests that around 30% of daily social media content generated by malicious programs or bots is misleading. Therefore, there is a pressing need to enhance the credibility of social networks through efficient detection of misinformation and bot-driven activities.

A bot refers to a software application designed to perform specific functions over a network, commonly called an internet or web bot. Bots generally automate routine and repetitive tasks, often operating at speeds beyond human capacity. They are frequently used for purposes such as data collection, scraping dynamic web content, and accessing server resources more efficiently than humans. Unfortunately, some bots are misused to generate and spread false information by rapidly disseminating vast quantities of data.

Misinformation, or fake news, refers to intentionally crafted false content aimed at capturing attention, earning advertising revenue, or swaying public opinion. These narratives seem credible but contain purposeful falsehoods intended to engage or mislead audiences. For instance, allegations of Russian hacking activities in Virginia or claims linking Emmanuel Macron's campaign funding to Saudi Arabia have been widely circulated online. The rapid dissemination of unverified information complicates distinguishing between factual and fabricated content. Thus, differentiating genuine news from false stories remains a significant challenge.

One method to combat the spread of false information is to analyze user profiles and evaluate their credibility using various computational techniques. Cybercriminals often create fake profiles to engage in offenses such as hacking, identity theft, unauthorized access control, malicious linking, and email spamming. These fraudulent profiles can be generated by either bots or humans, with bot-created accounts frequently targeting numerous social media users simultaneously. The swift distribution of unverified content or accounts without adequate validation mechanisms poses a severe problem for online platforms.

With the exponential growth of social media accounts, identity deception has become increasingly problematic. Malicious entities utilize fake profiles, created by both bots and human agents, for various harmful purposes. This system seeks to mitigate the issue by filtering out bot accounts during data preprocessing and subsequently using a Recurrent Neural Network (RNN) algorithm to classify human accounts as authentic or fraudulent, based on multiple parameters.

Literature Review

Naman Singh et al. [1] proposed a framework for identifying fake accounts on social media using diverse machine learning algorithms. The system creates training rules from fraudulent account data to classify profiles as genuine or fake. Physiological and statistical analysis techniques are used for feature extraction and model training. The framework adheres to social media privacy guidelines, avoiding the extraction of specific user details and instead utilizing aggregate profile data, validated against synthetic datasets for machine learning classification.

Sarah Khaled et al. [2] introduced a detection system for malicious social media accounts using an SVM classification algorithm. The method employs Support Vector Machine (SVM) training boundaries to develop a robust training module using a neural network variant called SVM NN. SVM is used for extracting microfeatures and selecting attributes, while NN constructs unique training modules. This hybrid approach improves detection accuracy while reducing computational overhead compared to conventional machine learning methods and ensemble classifiers.

FatihCagatayAkyon and M. EsatKalfaoglu [3] designed a solution for detecting fraudulent and automated Instagram accounts using soft computing algorithms combined with various preprocessing methods. Their approach uses descriptive statistics to generate classification features, facilitating efficient identification of fake and automated profiles. They also propose a cost-effective feature selection strategy using genetic algorithms to optimize classification accuracy. To handle data imbalances in fraudulent account datasets, the SMOTE-NC algorithm is employed, improving real-time dataset evaluation.

Sowmya P and Madhumita Chatterjee [4] developed a method to detect fake and duplicate Twitter accounts using classification and distance-based measurement techniques. They utilized statistical features with the C4.5 classification algorithm to identify cloned accounts, performing comparative analysis to evaluate performance. Results indicated that their clone detection method outperformed the C4.5 algorithm in terms of effectiveness. However, dependence on profile attributes for duplicate detection sometimes led to lower success rates. Nevertheless, the C4.5 decision tree provided reliable supervision for comprehensive detection efforts.

RanojoyBarua et al. [5] presented a deep learning system for identifying false news articles. Their method uses an ensemble classification approach, employing LSTM and GRU algorithms to determine whether an article is accurate or deceptive. Additionally, they developed an Android mobile application to verify news article authenticity. Various NLP techniques were applied to generate meaningful tokens for feature extraction and classification.

Mehmet SEVI and Ilhan AYDIN [6] proposed a system for detecting fraudulent Twitter accounts using classification and data augmentation techniques. They used algorithms such as Adjacent Neighbor, Random Forest, and Logistic Regression to classify fake accounts. The Random Forest method builds multiple decision trees, formulates helpful rules, and applies voting methods for evaluation, achieving higher detection accuracy compared to traditional classifiers.

Gaurav Shetty et al. [7] explored a system for emotion analysis and spam account detection on Twitter using machine learning. The system monitors user signup behaviors and identifies high-diffusion patterns among newly created accounts. It gathers user input and profile data for classification purposes.

Ngoc C. L. et al. [8] developed a framework for fake account detection using a hybrid machine learning model with a random walk approach. The system employs an empirical ranking method that integrates graph-based and feature-based strategies to identify malicious Facebook profiles. Classification is performed using SVM and Sybil Walk algorithms, tested on over 10,000 Facebook accounts to ensure accuracy.

To enhance efficiency and minimize computational complexity, several emerging machine learning models focus on leveraging domain-specific information. Both supervised and unsupervised machine learning techniques are widely used for identifying malicious profiles [9].

Another essential factor for evaluating system performance is the Twitter dataset used, along with metrics assessing the model's efficiency. Misinformation in the dataset can lead to overfitting during training, reducing the model's generalization capability [10]. Many researchers validate their models on extensive datasets from online social networks (OSNs), which typically include data from authentic users as well as a smaller segment of malicious entities, often labeled to indicate normal or harmful activities.

Proposed System Design

Current methods for detecting malicious activities or bots, as outlined in [12] and [18], still struggle with challenges such as high false alarm rates and low classification accuracy. Our system begins by collecting data from Twitter accounts via the Twitter API, extracting information from recently accessed tweets. Social media platforms frequently fail to detect bots or fake profiles effectively. To tackle this issue, we integrate NLP and machine learning techniques. Data is collected from various social media channels and stored in repositories and dataset files. Given that data from sources like

Twitter is often unstructured, it is crucial to preprocess it using specialized sampling and filtering methods.

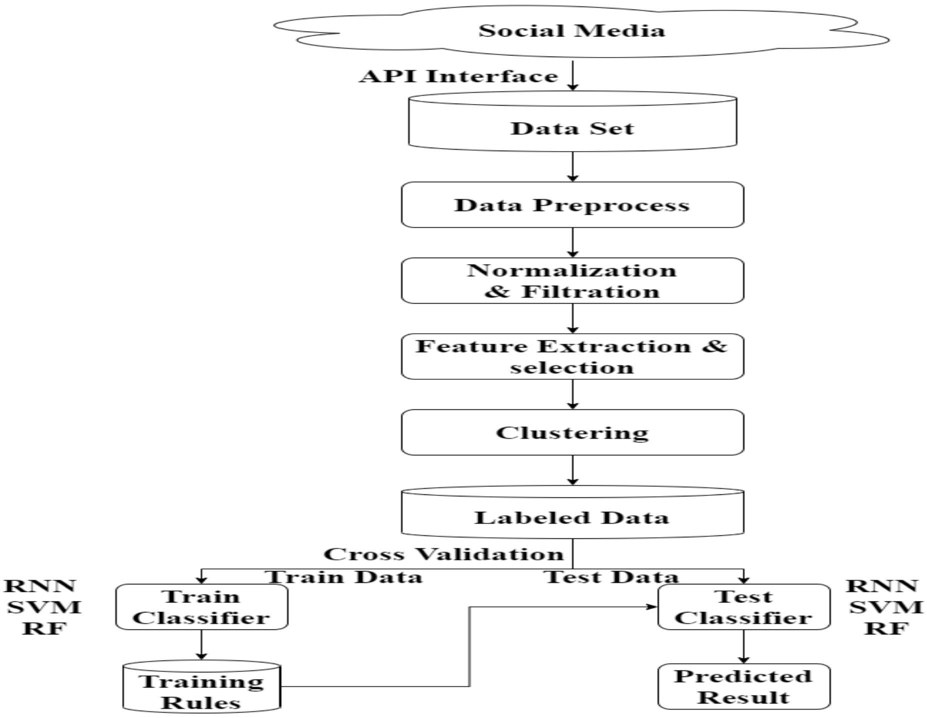

Figure 1: System architecture

We employ systematic sampling for data partitioning and use a Bloom filter to eliminate misclassified instances. Our preprocessing techniques include sentence segmentation and tokenization, with tokenized words stored in string vectors for efficient mapping. Natural Language Processing (NLP) methods, such as stop word elimination and lemmatization, are applied during the preprocessing stage. Features are extracted using techniques like TF-IDF and co-occurrence correlation, selected based on quality thresholds, and then fed into the classification algorithm. This procedure is carried out for both training and testing datasets. We apply a combination of Deep Learning and Machine Learning algorithms, including Support Vector Machine (SVM), Naïve Bayes (NB), Fuzzy Random Forest (RF), and Recurrent Neural Network (RNN), to perform classification. The system can detect and filter out suspicious entries in real time, achieving high accuracy for both synthetic and real-time data streams. It also demonstrates a low error rate and performs well on both homogeneous and heterogeneous datasets. Figure 1 provides a detailed overview of the system architecture.

Data Collection: The dataset for this study was obtained from Twitter using its Application Programming Interface (API). Each profile includes various attributes, such as name, username, number of followers, number of tweets, retweets, and more. These attributes form the basis for profile classification.
Preprocessing: Preprocessing involves removing stop words from sentences. Subsequently, lemmatization is used to find the root form of each word based on its context. Punctuation marks are

removed, and contractions like "can't" and "isn't" are expanded to "cannot" and "is not."
Filtering & Normalization: Data categorization involves removing misclassified records, null values, and invalid entries. This results in a balanced dataset, reducing the risk of overfitting during model training.

Feature Extraction and Selection: Features are extracted from the normalized dataset, including those derived from policy-based techniques. Each attribute represents profile data values and is selected according to predefined threshold criteria.

Clustering: In cases where class labels are unavailable, clustering is used to generate labels for the entire dataset. The labeled data is then used to train a classifier, establishing Background Knowledge (BK) based on binary class labels. Cross-validation is applied post-clustering to split the records into training and testing sets.

Classification: Supervised Machine Learning (SML) is used to classify the data using the labeled information. The Recurrent Neural Network (RNN) algorithm is utilized for deep learning to detect identity fraud on social networks. Multiple decision trees are generated with randomly selected features from the feature set, and the majority class output from these trees is considered the RNN result. The system's performance is evaluated using various metrics, such as accuracy, precision, recall, and F1-score.

Algorithm Design

Dataset description

Table 1: The dataset for the planned system was mined from Twitter consuming the Twitter- API

No	Attribute_ Name	Details
1	USERName	Display Name of the Twitter Account Holder
2	SCREENNAME	Username or nickname of the Account
3	FORMED	Date and time when the account was created
4	PROFILE IMAGE	Profile image of the user
5	LOCATION	User's location
6	CLIENT-LANGUAGE	Desiredprogramdesignated by the account holder

7	Any Friend_COUNT	Overallquantity of friends
8	GROUPS_COUNT	Total number of fans
9	STATUS_COUNT	Total measure of tweets made by the user
10	LIST_ COUNT	Quantity _of_twitter crowds the account belongs to
11	TWEET_ COUNT	Total quantity of tweets declared by the user
13	UTC_ OFFSET	Quantity_OF_Twitter clusters the account belongs to
14	User_longitude	USER_Longitude_cost of the user's location
15	LATITUDE	Latitude cost of the user's location
16	Twiter_Post	Past tweet made by the customer
17	User_link/url	User link/url of the user's account profile
18	DESCEXP_Record	Explanation of the dispatched data
19	Location for Time	Present location's time area { GMT }
20	Sex Type	account owner sex type
21	Client_Age	ClientAge of the account owner
22	Tweets	Number of tweets dispatched by the account owner
23	Title	Explanation of the tweet title
24	Re-tweet	Re-tweet topic name
25	Affirmative Sentiment	Number of Affirmative tweets
26	Destructive Sentiment	Number of Destructive tweets

Table-1 outlines the changes Data Set collected from Twitter accounts, containing approximately 26 attributes used for both Training and Testing. Everyelement represents aspects of an individual's user profile. Various cross authenticationmethods are applied to enhance the classification accuracy of real-time data.

System Testing Algorithm

This division provides a thorough overview of the systems employed in the execution. The primary process for detecting false and actualuniqueness on social-media is the Random Forest algorithm. This supervised machine learning technique uses labeled examples for training.

1: Random Forest

Input: Selected features for all test examples, denoted as d[(i...n)], along with training database guidelines T[(1)],...T[(n)]

Output: False vs. Actual accounts.

Steps:

1. For everyexample d[(i)] in d, select n attributes randomly from d[(i)] using the specified formula,

$$\text{Tree-set}[(\ k\ )] = \sum_{k=1}^{n} \text{Element}[d[(i)]k\ldots..d[(n)]n]$$

2. For each (t[(i)] into t)

$$\text{Train }[(m)] = \sum \text{Arribute }[t[(i)]k\ldots..\underset{k=1}{\overset{n}{t[(n)]}}n]$$

3. Compute weight amongst train and test instance

$$\text{Tree-set}[k].\text{weight} = \text{likeness }(\text{Tree-set}[(k)]\sum_{m=1}^{n}\ \text{train}[(m)]\ )1$$

4. If (Tree-set[(k)].weight>Th), then assign Tree-set[(k)].class to Train[(m)].class and exit

5. returnTree-set[(k)].class

2. Recurrent Neural Network (RNN)

PreparationPractice

Input: Preparation dataset Train-Data(), numerousinitiation functions(), and threshold Thr.

Output: Extracted Types from the Feature-set() for the finalized training component..

Step 1: Usual the input data block d[], the initialization function, and the age size., Step 2 :Types.pkl ← Extrac-t Types (d[])

Step 3 :Types-set[] ← optimize(Types.pkl)

Step 4 : Return Type_set[]

Testing Process

Input: Training dataset Test-DBLits[], training dataset Train-DBLits[], and threshold Th..

Output: Result-set<<class-name, Likeness_Weight>>contains all entries where the weight is greater than Th

Step 1: For every testing record, apply the following equation to process all data through the convolutional layer for both training and testing.

4o mini

$$testFeature(k) = \sum_{m=1}^{n} (featureSetA[i] \ldots \ldots .. A[n]TestDBLits)$$

Step 2: Create feature vector fromtestFeature(m) using below function.

Extracted_FeatureSet_x [t….…n] = $\sum_{x=1}^{n}(t)$□$testFeature$ (k)

The "Extracted FeatureSet_x[t]" denotes the output of pooling layers obtained from every convolutional layer and passed to the subsequent convolutional layer. This layer grasps the features take out from each case in the testing dataset.

Step 3: For each training example, use the function below.

$$trainFeature(l) = \sum_{m=1}^{n} (.featureSetA[i] \ldots \ldots .. A[n]TrainDBList)$$

Step 4: Create a new feature vector from trainFeature(m) using the function below.

.Extracted_FeatureSet_Y[t……n] = $\sum_{x=1}^{n}(t)$□$TrainFeature$ (l)

Extracted_t" is the result of everycombining layer extract from every convolutional layer and forwarded toward the subsequent convolutional level. This layer contains the extract features of each oneexample within the preparation dataset.

Step 5 :Now, estimate each test account using the complete exercise dataset in the solid layer.

$$weight = calcSim\left(FeatureSetx \vee \sum_{i=1}^{n} FeatureSety[y] \right)$$

Step 6: Return Weight

Results and Discussions

To evaluate the system's performance, accuracy metrics were calculated. The system is built on a Java 3-tier architecture framework, running on an INTEL I3 processor at 2.8 GHz with 4 GB of RAM, and operates within an open-source environment.

After implementing the system, a comparison was made between several existing systems and the proposed solution. We assessed the performance of ReLU on the Twitter dataset, carrying out parallel experiments using different cross-validation techniques. The findings are presented in Table 2.

Our analysis indicates that 10-fold cross-validation yielded the highest classification accuracy, achieving 95.30% and 96.10% for the RNN model.

Table 2: Classification correctness using the confusion matrix for RNN (ReLU).

RNN (Sigmoid)	Fold_5	Fold_10	Fold_15
Accuracy	94.20	95.30	96.10
Precision	94.30	95.70	95.30
Recall	94.15	95.80	96.40
F1 Score	93.20	95.60	96.50

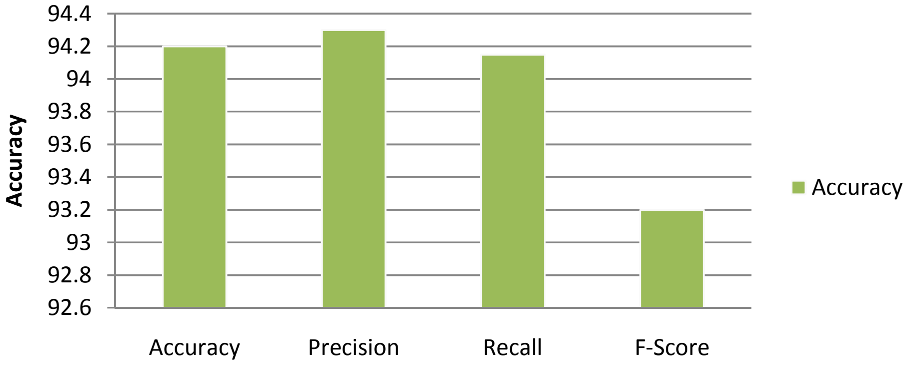

Figure 2 : Accuracy of detecting fake accounts with RNN using 5-fold cross-validation.

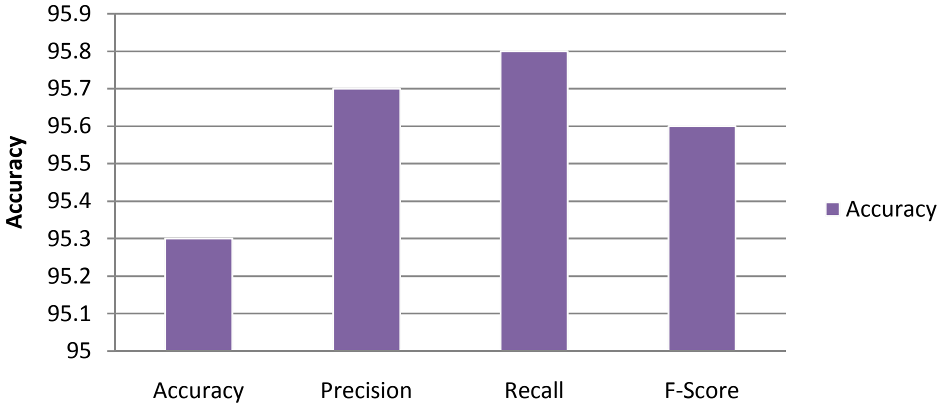

Figure 3 : Accuracy of fake account detection consuming RNN with 10-fold cross-validation.

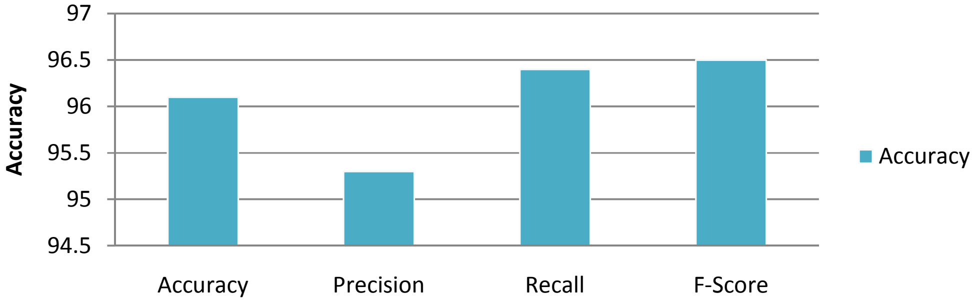

Figure 4 : Accuracy of fake account exposure using RNN with 15-fold cross-validation.

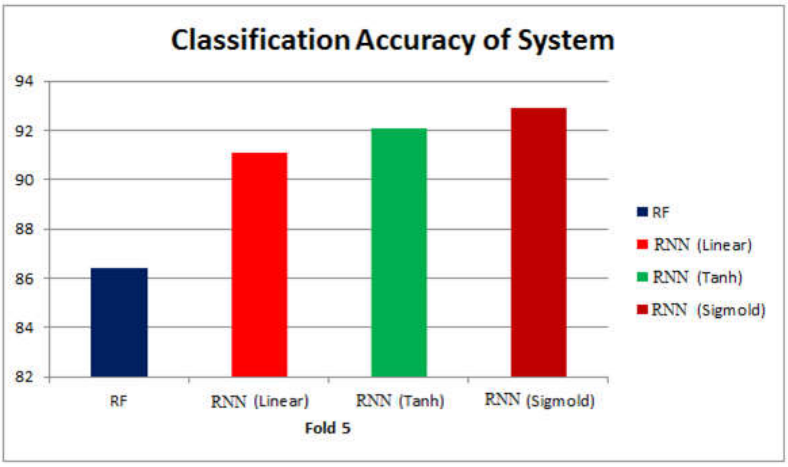

Figure 5: Result Graph for variant in accuracy for RF, RNN (ReLU), RNN (Tanh) and RNN (Sigmoid) using 5 fold cross validation.

We conducted a performance analysis using seven algorithms, including Random Forest, Naïve Bayes, and the proposed RNN algorithm. The performance assessment utilized a confusion matrix. The RNN algorithm was tested with three activation functions: SIGMOID, TANH, and ReLU. Overall, the RNN demonstrated higher detection accuracy than other machine learning classifiers, with ReLU outperforming Sigmoid and TANH. Additionally, the system addresses the detection of fake profiles from compromised accounts using machine learning techniques. The confusion matrix provided an evaluation of overall accuracy, including correct and incorrect classifications, as well as precision and recall metrics. The findings indicate that the proposed system achieves better accuracy compared to existing methods.

Conclusion and Future Work:

The future approach aims to detect fake accounts on social media platforms using Machine Learning and Deep Learning algorithms. Techniques such as SVM, Random Forest (RF), Naïve Bayes (NB), and RNN are utilized to address the challenge of identifying fraudulent profiles. Among these, the RNN model with ReLU activation function delivers the highest performance, achieving 94.7% accuracy. The system's effectiveness depends on the choice of classification technique and the dataset used. Experimental results show that the proposed method provides favorable outcomes compared to the current state-of-the-art techniques discussed in the literature. The use of the RNN classifier has significantly improved classification accuracy relative to other machine learning approaches.

The system's performance has been enhanced by employing different activation functions on the same dataset, eliminating the need for manual classification of fake profiles, which is both time-consuming and resource-intensive. Our approach offers a novel solution for identifying fake accounts on Online Social Networks (OSNs). Despite recent advancements improving detection efficiency, the system's performance can still be affected by unpredictable factors. In our analysis, we increased prediction accuracy through techniques such as preprocessing, data sampling, data partitioning, and model training and testing. The proposed method delivers faster and more precise results. Although data was collected from a limited number of social media platforms, future research could enhance detection capabilities by incorporating additional parameters, such as user-shared metadata and the credibility of shared content, which are not currently considered.

REFERENCES

1. Drouin, Michelle, Daniel Miller, Shaun MJ Wehle, and Elisa Hernandez. Why do people lie online?“Because everyone lies on the internet, Computers in Human Behavior 64 (2016): 134-142.
2. Jupe, Louise Marie, AldertVrij, GalitNahari, Sharon Leal, and Samantha Ann Mann. The lies we live: Using the verifiability approach to detect lying about occupation., Journal of Articles in Support of the Null Hypothesis 13, no. 1 (2016): 1-13.
3. Li, Yixuan, Oscar Martinez, Xing Chen, Yi Li, and John E. Hopcroft. In a world that counts: Clustering and detecting fake social engagement at scale., In Proceedings of the 25th International Conference on World Wide Web, pp. 111-120. International World Wide Web Conferences Steering Committee, 2016.
4. Tuna, Tayfun, EsraAkbas, Ahmet Aksoy, Muhammed Abdullah Canbaz, UmitKarabiyik, Bilal Gonen, and RamazanAygun.,User characterization for online social networks. ,Social Network Analysis and Mining 6, no. 1 (2016): 104.
5. Galan-Garcia, Patxi, Jose Gaviria de la Puerta, Carlos Laorden Gomez, Igor Santos, and Pablo GarcíaBringas. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. Logic Journal of the IGPL 24, no. 1 (2016): 42-53.
6. Stanton, Kasey, Stephanie Ellickson-Larew, and David Watson. Development and validation of a measure of online deception and intimacy., Personality and Individual Differences 88 (2016): 187-196.
7. Kim, Jihyun, and Howon Kim., Classification performance using gated recurrent unit recurrent neural network on energy disaggregation., In 2016 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 105-110. IEEE, 2016.
8. Zhang, Yong, MengJooEr, RajasekarVenkatesan, Ning Wang, and MahardhikaPratama. Sentiment classification using comprehensive attention recurrent models., In 2016 International joint conference on neural networks (IJCNN), pp. 1562-1569. IEEE, 2016.
9. Peddinti, Sai Teja, Keith W. Ross, and Justin Cappos. Mining Anonymity: Identifying Sensitive Accounts on Twitter.,arXiv preprint arXiv:1702.00164 (2017).
10. Dimpas, Philogene Kyle, Royce Vincent Po, and Mary Jane Sabellano. Filipino and english clickbait detection using a long short-term memory recurrent neural network., In 2017 International Conference on Asian Language Processing (IALP), pp. 276-280. IEEE, 2017.
11. Rajesh Purohit Bharat SampatraoBorkar ,Identification of Fake vs. Real Identities on Social Media using Random Forest and Deep Convolutional Neural Network, in International Journal of Engineering and Advanced Technology, Issue-1 7347-7351 IJEAT 2019
12. B.PanduRanga Raju, B.Vijaya Lakshmi, C.V.Lakshmi Narayana, Detection of Multi-Class Website URLs Using Machine Learning Algorithms, *International Journal of Advanced Trends in Computer Science and Engineering,* pp. 1704-1712 ,Volume 9, No.2, 2020.