COMPUTOLOGY: Journal of Applied Computer Science and Intelligent Technologies
Volume 4, Issue 1, Jan-Jun 2024, pp. 1-19
DOI: 10.17492/computology.v4i1.2401
https://www.journalpressindia.com/cjacsit

Fake Profile Detection and Stalking Prediction on X using Random Forest and Deep Convolutional Neural Networks

Baribor Deedee, Taylor Onate** and Victor Emmah****

ABSTRACT

This study employs Random Forest (RF) and Deep Convolutional Neural Networks (DCNN) to predict stalking behavior on X and detect phony profiles. The source of the dataset was Kaggle. The model was developed and evaluated using the Object Oriented Analysis and Design (OOAD) methodology. Utilizing the Python computer language, the RF&DCNN algorithms were implemented. Real-time detection and prediction are provided by the algorithms, which process the input data iteratively and update the model parameters in response to fresh observations. Statuses_count, followers_count, friends_count, favorites_count, and listed_count are among the input parameters provided into the model. By including these parameters in the model, profiles can be predicted effectively and with accuracy. Based on the research, an accuracy level of 93.89% with an error rate of 6.104 was achieved. With an accuracy rate of 86.57% and an error rate of 13.43%, the proposed model outperformed the current one in terms of effectiveness. The outcomes show how well the RF and DCNN based prediction model works to identify fake profiles and predict stalking. By putting out a novel method for identifying phony profiles and forecasting stalking utilizing RF and DCNN, this study advances the field of anomaly detection operations.

Keywords: *Fake profile, X, Stalking, Machine Learning, RF classifier, DCNN classifier.*

1.0 Introduction

People all over the world today rely on social media for a variety of reasons, including searching for resources and information, sharing thoughts, experiences, and knowledge, and increasing their social connectivity.

**Corresponding author; Lecturer, Department of Computer Science, Rivers State Polytechnic, Bori, Rivers State, Nigeria (E-mail: baribordeedee@yahoo.com)*
***Lecturer, Department of Computer Science, Rivers State University, Port Harcourt, Rivers State, Nigeria (E-mail: taylor.onate@ust.edu.ng)*
****Lecturer, Department of Computer Science, Rivers State University, Port Harcourt, Rivers State, Nigeria (E-mail: victor.emmah@ust.edu.ng)*

However, the same characteristics that make social media useful for consumers also leave them vulnerable to many forms of online fraud. Offenders prefer to use fake personas to carry out their nefarious intentions (Secchiero, 2012).

False identities can be used for a variety of nefarious purposes, such as spreading malware, tricking consumers into visiting dangerous websites, obtaining credentials via fabricating messages, influencing users' behavior, cyberbullying, falsifying an account's creditworthiness: For instance, fabricating personas using the name of a well-known somebody and sharing unsolicited content on it with the intention of discrediting that individual and skewing perceptions to give the impression that a product is superior to its rivals', false identities are employed to generate phony likes for advertisements of that product (El-Azab *et al., 2016*).

Both people and machines (bots) are capable of creating fake profiles. X (Twitter) is among the most widely used social media platforms. X is the subject of several nefarious acts, including impersonating others online, making up profiles in order to stalk them, cyberharassment, and other online provocations that compromise privacy and damage X's reputation on social media platforms (Prathyusha *et al.,* 2021).

Identifying these phony profiles is one of the difficult issues in social network security (Huang *et al,* 2021). Because of this, cybersecurity tools and safeguards are now required to prevent people from engaging in cyberbullying, such as stalking others using fictitious personas. This study presents a framework for applying machine learning approaches to classify an X profile as authentic or fraudulent. Random Forest & Deep Convolutional Neural Networks is suggested, and the prediction of stalking will employ the same approach.

2.0 Related Work

Balakrishnan *et al.,* (2020), developed a system for automatically identifying cyberbullying in Twitter tweets using machine learning techniques. This technique classifies the tweets into four categories: legitimate tweets containing psychological characters, sentiment, and feelings; aggressor tweets; bully tweets; and spammer tweets. 5453 tweets made up the dataset used for the experiment, which was carried out using the Naïve Bayes and J48 Machine Learning algorithms.

Mohammed (2020), offered a novel method that serves two purposes: it uses Semantic Web Rule Language (SWRL) rules and ontology engineering to categorize and identify Twitter bots. Identifying the characteristics that set a bogus account (bot) apart from the actual one and classifying bogus accounts as spam bots or false followers through

inductive learning, the authors used Web Ontology Language (OWL), Semantic Web Rule Language (SWRL) rules, and reasoners. The authors claim that their method could correctly identify the phony account with 97% accuracy in the first step. After then, 94.9% of the fraudulent accounts were correctly classified as spam or false follower bots. Furthermore, the ontology classifier has been found to be a more interpretable model with more straightforward decision criteria than other machine learning classifiers.

Eastee & Jan (2018) used Random Forest, Adaboost, and Support Vector Machines to classify Real from Fake Twitter profiles. They discovered that Random Forest produces the best results. It has been noted that social media platforms offer a plethora of characteristics that serve to characterize the identity of individual profiles. Location, name, profile picture, number of friends and followers, account creation date, number of URLs, number of status updates, and number of retweets are a few examples.

Fatih & Esat (2019), the identification of bogus and automated records that result in a phony Instagram joint effort is covered in their work, Instagram Fake and Automated Account Detection. In addition to a cost-sensitive element extraction method based on a genetic computation for choosing the best attributes for computerized account characterization, this study presented an anticipated work for erroneous and robotized account recognition. A few AI calculations, including Guileless Bayes, strategic relapse, support vector machines, and neural networks, were used to discriminate between real and artificial accounts. The best scores were obtained by neural organizations and SVM. The neural organization received the highest score of 95%, while SVM received 86%.

Reza & Soheila (2020), employed a multi-objective hybrid feature selection method to identify fraudulent accounts, which aids in feature set selection with the best classification performance. The candidate feature set was first determined by using the Minimum Redundancy – Maximum Relevance algorithm (mRMR) to the features with the lowest redundancy and the strongest relationship to the target class. The stable feature set with the fewest features that may yet achieve optimal performance is then selected as the final feature set for the detection operations. Two Twitter datasets were used to assess the proposed strategy, and the results were compared to those of other well-researched and successful methods. The outcomes demonstrate that the recommended classifier strategy outperforms the available methods.

Saberi *et al.* (2007)**.** Here, the writers offered an ensemble approach for identifying phishing frauds. Spam and non-spam are categorized using data mining classification techniques including Naive Bayes, K-nearest neighbor, Poisson probability theory, and Naive Bayes. To achieve greater accuracy, the output of various classifiers was combined. Accuracy ratings for the Naive Bayes, k-nearest neighbor, and Poisson algorithms are

88%, 87.5%, and 90.6%, respectively. With the combination of these three methods, accuracy is raised to 94.4%. Other techniques like SVM and neural networks can help increase the accuracy of detecting scams.

Ratkiewicz *et al.* (2011) presented a framework for identifying the spread of political misinformation. The political misinformation propagated by hackers on Twitter was identified by the authors using machine learning techniques. To assess user behavior, this framework blends crowdsourced and topological content-based features. AdaBoost and SVM, two classifiers, were employed to generate the output. Both classifiers—with and without resampling are applied. AdaBoost and SVM without Re-sampling have accuracy percentages of 92.6% and 88.3%, respectively.

Egele *et al.* (2015) introduced the COMPA technology, which finds compromised social network accounts. The way people behave on social networks serves as the basis for this system. Normal user behavior is constant, whereas COMPA notices compromised accounts that exhibit more erratic activity. The behavioral profile in COMPA is created by using the previous message that the account has sent. The behavioral profile is compared whenever a new message is created. COMPA marks a message as compromised if it differs from the original with a behavioral profile. This method yields positive outcomes when used on Facebook and Twitter. Facebook's false positive rate is 3.6%, whereas Twitter's is 4%.

Saeid (2020) suggested what he deemed an effective technique for spotting phony Instagram accounts in this study. First, a dataset of real and phony accounts was assembled for the model that was given. Next, in order to identify phony users on the dataset, the bagging classifier was trained using the gathered dataset as input. To further assess the method's efficacy, the suggested approach was further tested in terms of classification accuracy against five popular machine-learning classifiers. The experimental findings demonstrate that the suggested strategy outperforms other algorithms taken into consideration, accurately categorizing with a low mistake rate more than 98% of the accounts.

Yazan (2015) developed a method to stop fraudulent accounts on social media networks. Facebook and Tuenti real-time data sets were gathered by the author. Using a feature set, this method finds phony accounts. Fourteen features were taken from twenty datasets, while the eighteen features of Facebook were taken from the Facebook data collection. The data was classified using the Random Forest classifier. A Random Forest was used to create 500 decision trees for the Tuneti data set and 450 decision trees for the Facebook data set during the training phase. Thirteen features from Tuenti and three from

Facebook are randomly selected by the decision tree out of a total of fourteen features. SVM and Naive Bayes algorithms were also used.

Ten-fold cross validation is used in this procedure. The AUC of 0.7 for Facebook and 0.76 for Tuenti is provided by the random forest. For Facebook, Navie Bayes yields an AUC of 0.63 and SVM yields an AUC of 0.57. For tuneti, the AUCs from SVM and Naive Bayes are 0.59 and 0.64, respectively.

Asante *et al*., (2021), suggested a technological approach for content-based cyberstalking detection. The proposed methodology made use of a few modules: evidence, content detection, filtering, identification of messages, and detection (content and offender profiling). The authors used machine learning, data mining techniques, digital forensics, and profiling to examine text, picture, and media material, collect evidence, and accurately profile offenders.

Gayatri *et al.,* (2020), recommended machine learning-based methods to spot fraudulent accounts that could trick consumers. The dataset developed specifically for this purpose was preprocessed, and then machine learning techniques were applied to detect fake accounts. To find fraudulent accounts, algorithms including support vector machines, decision trees, and logistic regression were used. After comparing the categorization success of different approaches, it was established that logistic regression produces better outcomes.

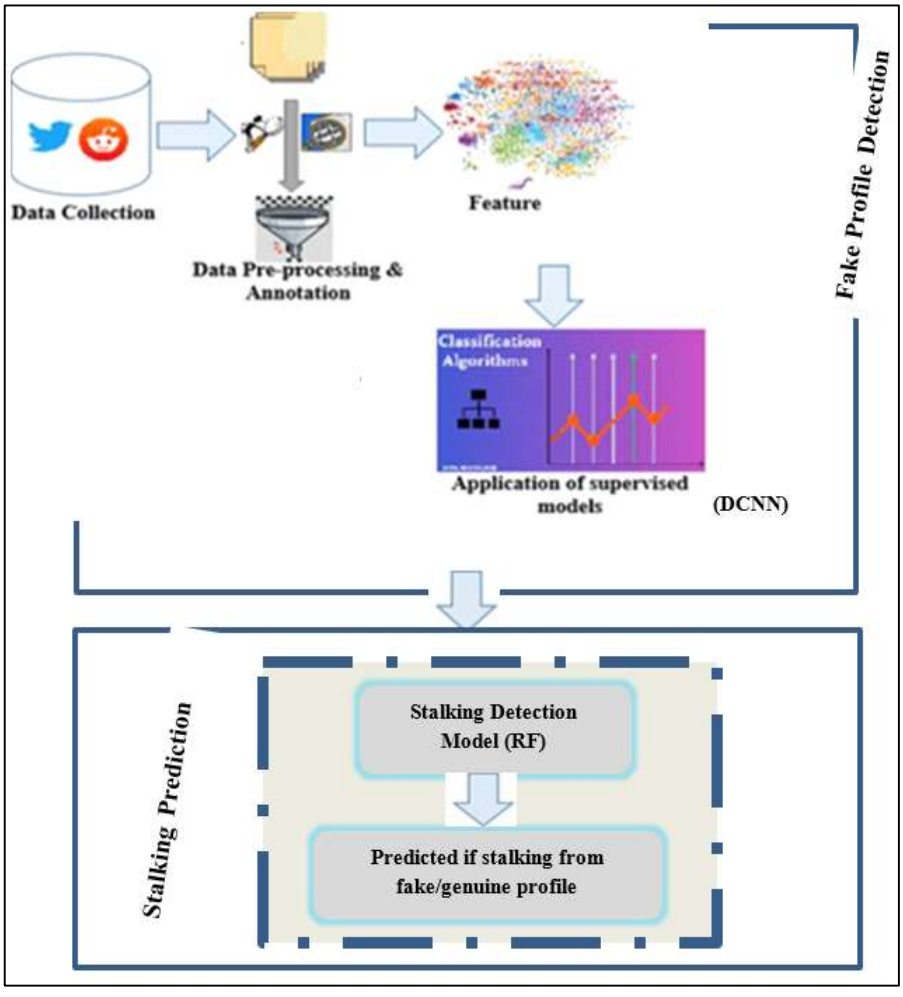
Chakraborty *et al.,* (2022), in their paper entitled Machine Learning Techniques for Fake Profile Detection. By utilizing a range of machine learning approaches, the authors' suggested detection model can differentiate between authentic and fraudulent Twitter profiles based on several parameters like the number of followers and friends, status updates, and more. Using the Twitter profile information, the authors classified phony accounts as TWT, FSF, and INT and authentic accounts as TFP and E13. The writers cover LSTM, XG Boost, Random Forest, and Neural Networks as machine learning algorithms in their work. The scientists concluded that XG Boost, with its accuracy rate of 99.6%, is the greatest machine learning technique for detecting fake accounts on social networking networks.

Bhosale & Mane (2024) suggested a hybrid methodology to identify phony social media profiles. An analysis of the machine learning algorithms Naïve Bayes, Random Forest, AdaBoost, Support Vector Machines (SVM), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a new hybrid model that combines LSTM and GRU is conducted by the authors of Enhancing User Trust: A New Hybrid Model for Online Social Network Fake Profile Detection. The writers attained a 98.7% accuracy level using their hybrid model.

3.0 Design Methodology

The strategies and procedures employed for the data collecting are covered in this session. Figure 1 depicts the proposed system's system architecture.

Figure 1: Architectural Design of the Proposed System



There are two steps to the suggested work:

- Fake profile detection
- Stalking prediction

3.1 Fake profile detection

- The first step in the detection procedure is choosing the profile that has to be tested.
- After the profile has been chosen, the relevant feature(s) is/are selected, and the classification algorithm is applied.
- The trained classifier receives the extracted attributes. Every time new training data is introduced into the classifier, it undergoes regular training.
- The classifier establishes if the profile is authentic or fraudulent.
- Since the classifier might not have classified the profile with 100% accuracy, it receives feedback on its performance.
- As time goes on, more training data are collected, increasing the classifier's accuracy in identifying phony profiles. This procedure is repeated.

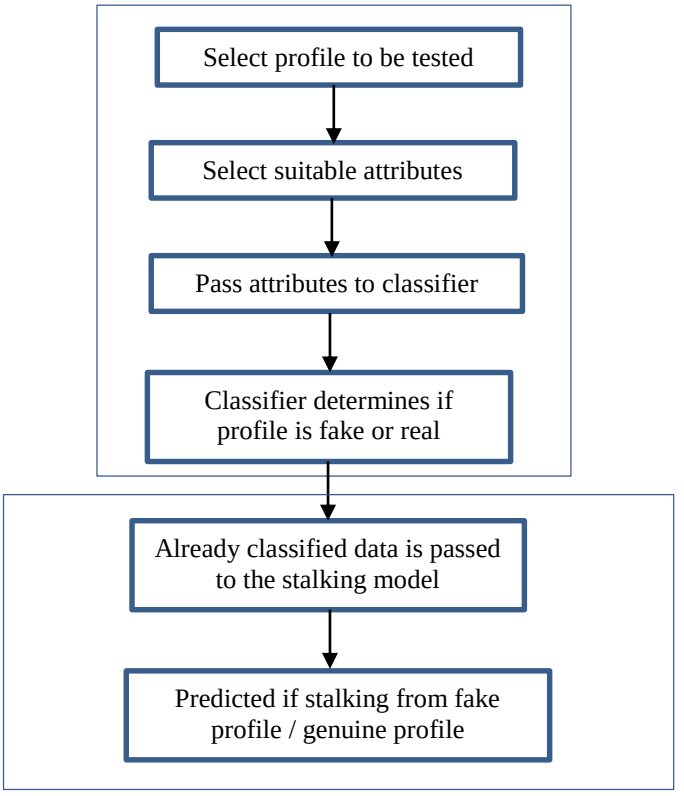
3.2 Stalking prediction

- Choosing the profile that requires testing is the first step in the detection process.
- Following the profile selection, the appropriate attributes, or features, are chosen, and the classification method is then applied.
- The trained classifier receives the extracted attributes. Every time new training data is introduced into the classifier, it undergoes regular training.
- The classifier establishes if the profile is authentic or fraudulent.
- Since the classifier might not have classified the profile with 100% accuracy, it receives feedback on its performance.
- As time goes on, more training data are collected, increasing the classifier's accuracy in identifying phony profiles. This procedure is repeated.
- The stalker model is loaded with the previously classified data.
- The classifier ascertains whether the individual is pursuing leads from a real or fictitious profile as represented in Figure 2.

3.3 Data collection

Data collection is required for the model to function. The dataset can be generated with Crawler and gathered from a variety of web resources. We have gathered two datasets via the internet from reputable sites, GitHub and Kaggle. However, we worked with a dataset that Kaggle obtained, and we used two CSV files one for fictitious users and the other for real users.

Figure 2: Stalking Prediction Flowchart



3.4 Data pre-processing

It is common for a dataset collected from many social media sites and other web apps to contain unique, unnecessary text or characters. For the Machine Learning classifier in the detection phase, clean and prepared data are required prior to assessing the machine learning algorithms. The datasets' data are standardized to a predefined format and filtered using keywords. During the pre-processing stage, natural language processing, or NLP, is typically utilized for several tasks. Data pre-processing techniques include stop word removal tasks, noise reduction, normalization, tokenization, stemming, and lemmatization. After the natural language processing pre-processing step is finished, clean data are sent to the next phase for feature extraction.

3.5 Feature extraction

The features utilized for machine learning algorithm performance evaluation and training are described in detail in this section. Just five of the 33 attributes that are available

to each user were selected in order to categorize them as malevolent or legitimate. Most of the remaining attributes are meaningless because they contain non-numerical data, such as the URL of the profile photo or the background color that is being used. Without any additional calculations, these are the public features that were obtained straight from the Twitter API:

- *statuses_count*: The total amount of tweets that a user has posted.
- *followers_count*: The quantity of followers an individual possesses.
- *friends_count*: How many buddies every user has. The people that this account is following are considered friends.
- *favourites_count*: Total amount of tweets favorited by a specific user.
- *listed_count*: The number of lists to which an account on Twitter is subscribed.

3.6 Algorithms used

3.6.1 Random forest

- **Step 1:** To start, select random samples from a given dataset.
- **Step 2:** After that, a decision tree using this method will be constructed for each sample. It will then retrieve the forecast outcome for each decision tree.
- **Step 3:** Select the forecasted result that garnered the greatest number of votes as the winner in the end.
- **Step 4:** Choose the predicted result that got the most votes to be the winner in the end.

Mathematically, Random Forest is given by:

$$Random\ Forest = \frac{1}{N}\sum_{I=1}^{N}(Fi - yi)2 \qquad \text{eqn(i)}$$

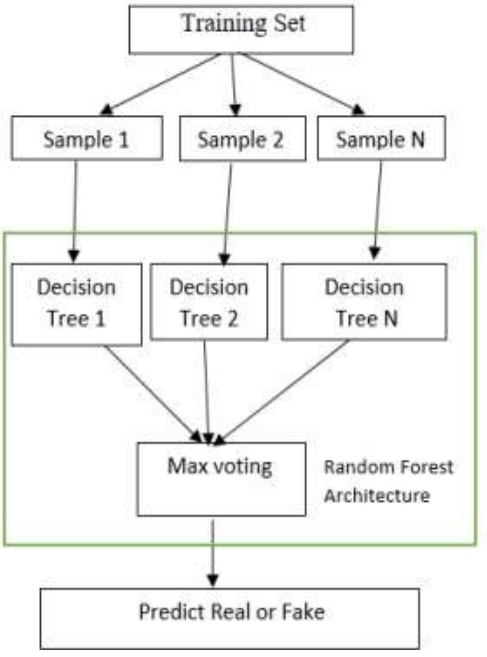
where N is the number of characteristics used to find comparable accounts, Fi is the value returned by X (Twitter), and yi is the original value used for feature i.

In the model, we choose to use the Random Forest machine learning technique in order to attain high accuracy. This decision was made because the random forest method yields results with a high degree of accuracy and can effectively handle missing values in the data. The architectural design of Random Forest algorithm is depicted in Figure 3.

3.6.2 Deep convolutional neural networks

Step1: Input Layer: The raw input data, which is usually given as a token sequence, is fed into the CNN's input layer.

Figure 3: Random Forest Architecture (Harish, 2023)



Step 2: Convolution Layers: The central part of a CNN is the convolutional layer. It is made up of several filters, or kernels, that convolve over the sequence that is input. Every filter extracts local features by summing and multiplying elements-wise between the filter weights and a subset of the input. This procedure aids in identifying various features and trends within the dataset.

Step3: Activation Function: The generated feature maps are subjected, element-by-element, to an activation function (ReLU, or Rectified Linear Unit) following the convolution process. The network may learn intricate correlations in the dataset. The activation function introduce non-linearities into the network.

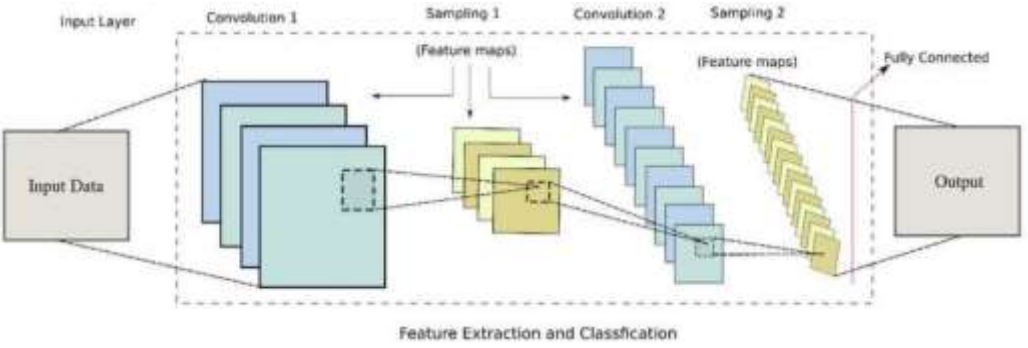
Step4: Pooling Layer: The most significant information is retained in the feature maps while the spatial dimensions are decreased by the pooling layer. It increases the network's resilience to minute spatial fluctuations and aids in the reduction of computing complexity.

Step 5: Repeat Steps 2-4 (convolution, activation, and pooling) are usually carried out several times to build a deep network. Through the combination of features learnt in prior layers, deeper layers are able to learn higher-level representations of the incoming data.

Step 6: Fully Connected Layer: The output is typically vectorized after multiple convolutional and pooling layers and sent to one or more fully connected layers. These layers, which resemble those in a conventional neural network, are in charge of prediction based on the features that have been extracted.

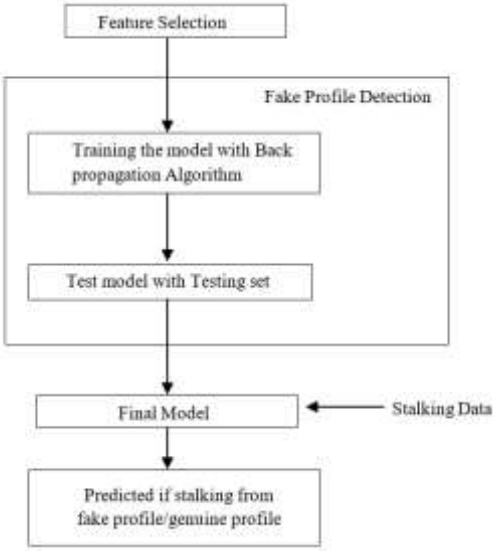
Step 7: Output Layer: The final predictions are generated by the CNN's output layer. The architectural design of DCNN is shown in Figure 4 below while Figure 5 represents the system's flowchart.

Figure 4: DCNN Architecture



Source: https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9)

Figure 5: Dataflow Diagram of the Proposed System



3.7 Extracted features used for classification

Table 1: Fake Profile's Detection Dataset (Beatriche, 2018)

Features(Name)	Real profile	Fake profile
statuses_count	The amount of tweeted status updates is higher on real profile behavior.	False accounts don't tweet frequently.
followers_count	Genuine users have a large following.	There is a decreased likelihood of spammers on profiles with thousands of followers.
friends_count	Real profiles typically have a large number of followers.	The friends of spammers are few in number.
favourites_count	More tweets are favorited by real users.	Fewer tweets are marked as favorites by a fake account.
listed_count	Having a large number of lists is normal for genuine users.	Almost never do fake users belong to lists.
FFratio	Reduced ratio levels indicate genuine users.	Fake accounts are more likely to have a higher ratio.
URLratio	Genuine profiles don't tweet URLs too frequently.	It's conceivable that bots will tweet links.
average_mentions	Genuine accounts employ more mentions in their tweets.	Less is mentioned in tweets by fake accounts.
average_hashtags	True users frequently include hashtags in their tweets	Real users have more hashtags than fake users do.
average_favorites	Reduced value demonstrates actual user	Higher value indicates malevolent user
average_retweet	Tweets that are not frequently retweeted are typically retweeted by real persons.	False users retweet additional well-known tweets
average_reply	Less people are responding to tweets from real profiles.	More replies are made to tweets by the fake account.

3.8 Dataset

A publicly accessible data collection was used in the design and successful testing of the suggested model. There are 1329 phony users and 1469 real users in the dataset as shown in Figure 6. The dataset has the following attributes: statuses_count, followers_count, friends_count, favorites_count, and listed_count. Within the dataset, data for testing and training are segregated. Classification algorithms are trained on a training dataset, and their efficacy is evaluated on a testing dataset. Twenty percent of the dataset is used for testing and eighty percent is used for training. Table 1 above shows fake profile detection dataset.

4.0 Implementation/Results

This system proposes a unified framework that uses a mix of two models Random Forest and Deep Convolutional Neural Network to classify an X (Twitter) profile as authentic or false. The same framework is also used to predict stalker behavior. The method begins by;

1. In order to use classification methods, features are chosen. For instance, gender, friendship count, status count, etc.
2. The model is trained using a data set of profiles that have previously been determined to be real or fraudulent, following the selection of the attributes.
3. A subset of the profile's attributes are taken out for classification.
4. Of this data set, 20% is used for testing and 80% is used for training.
5. The classification model receives the training data set. It is anticipated that it will correctly classify the testing data set after learning from the data collection.
6. The trained classifier is left to make the ultimate decision after the test set labels are removed.
7. A list of the most active stalkers will be compiled. These profiles will be regarded as the ones that need to be determined whether they are real or fraudulent.
8. Afterwards, valuable characteristics will be extracted and sent to the trained classifier.

Figure 6: Real and Fake Datasets

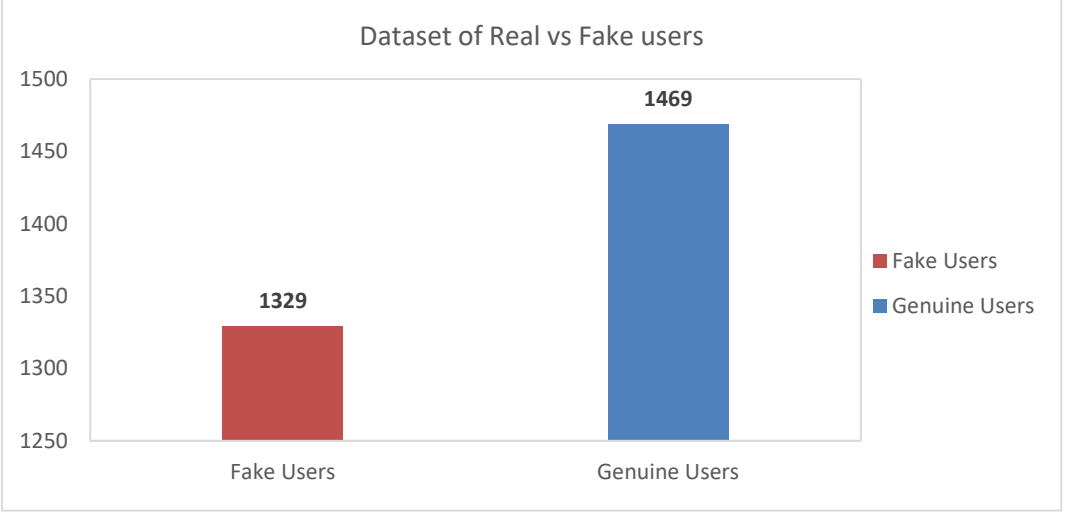


Figure 7: Graphical Analysis of Training Accuracy Vs Epoch

Figure 8: Graphical Analysis of Loss vs Epoch

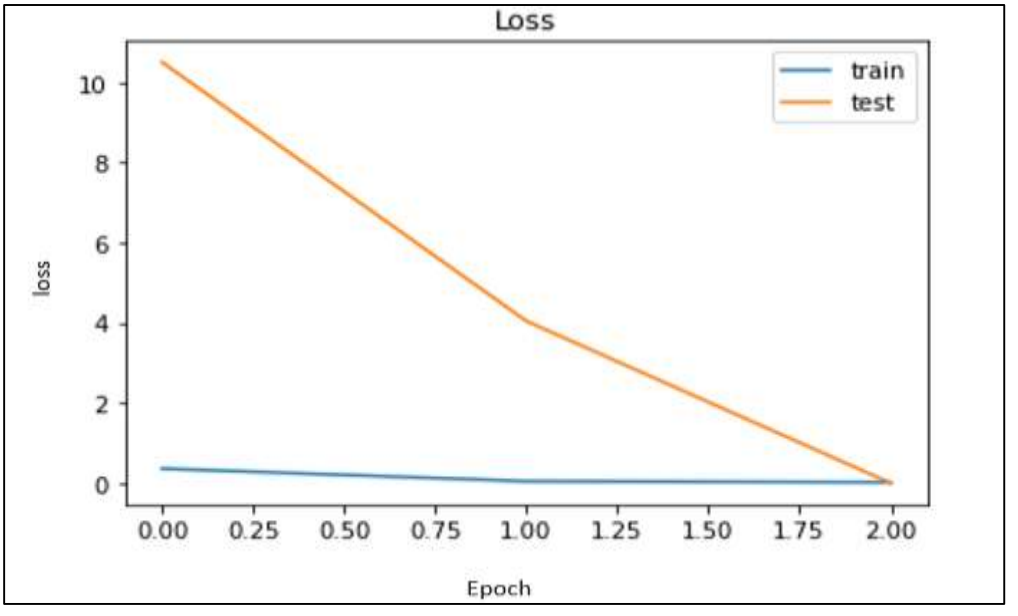


Figure 9: Predicted Stalking Result

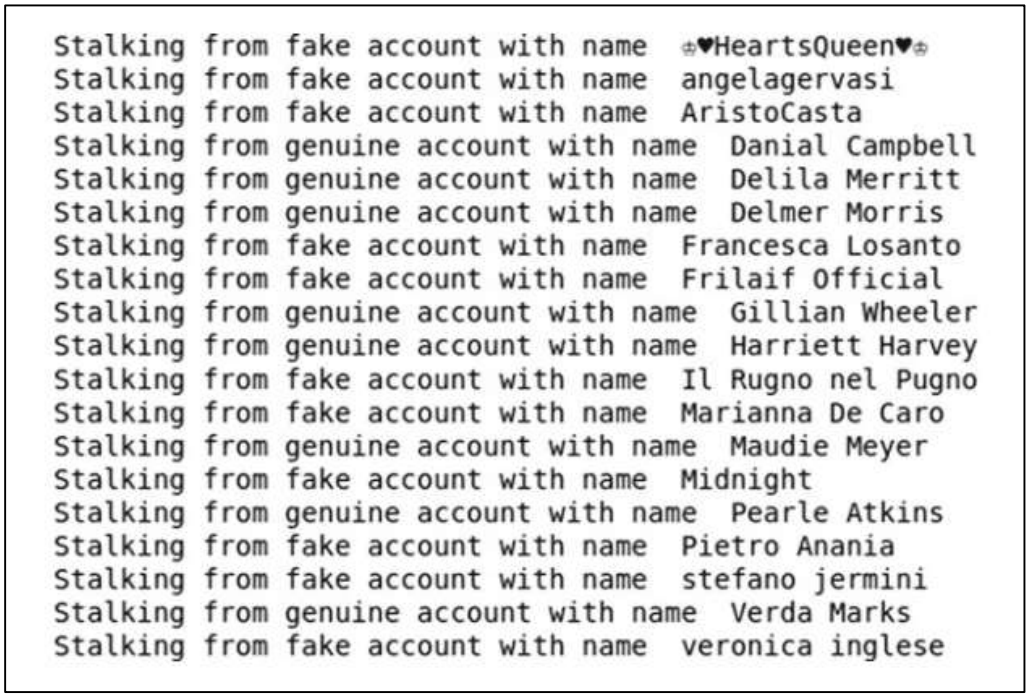


Figure 10: Classification Results

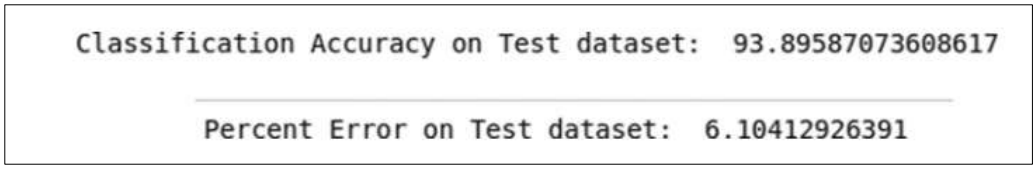


Figure 11: Results

	precision	recall	f1-score	support
Fake	0.92	0.96	0.94	275
Genuine	0.96	0.92	0.94	282
avg / total	0.94	0.94	0.94	557

5.0 Discussion

Python was used to implement the created model for identifying false profiles and forecasting stalking. A publicly accessible data collection was used in the design and successful testing of the suggested model. There are 1329 phony users and 1469 real users in the data set. Initially, the identified the features that were most likely to be involved in both identifying similar phony accounts and forecasting Twitter stalking. These characteristics directly affect the decision-making process when comparing the accounts' similarities. With an accuracy level of 0.93, the model as a whole yielded remarkable results much better than most other methods. Eighty percent of the data are in the training dataset and twenty percent are in the testing dataset.

To prevent overfitting, the classifier is well-trained using a larger amount of data during the training phase.

The classifier's performance metrics in the studies were support, F1 Score, accuracy, precision, recall, and recall. The experiment's outcome, as illustrated in Figure 11, indicates a predicted accuracy level of 93.89% with an error rate of 6.10%. As demonstrated by the results in Figure 11, the model is able to classify the dataset with an accuracy of 0.938958, a precision of 0.92 for phony profiles and a precision of 0.96 for real users, a recall of 0.92 and 0.96 for real and phony profiles, respectively, and an F1-Measure of 0.94 for both sorts of users. Figure 10 therefore shows profiles that are either stalking from a genuine profile or from a fake profile.

It makes sense that the ability of the classifier to refuse to classify a negative sample as positive corresponds to its precision. Recall is the classifier's natural ability to find every positive sample. The F1-score is the recall and precision weighted average. The quantity of genuine response samples that are present in the classroom is the final measure of support. The stalker prediction result is displayed in Figure 9. The user will be able to see the details by printing the account name, whether it is real or not. The model's training Accuracy Vs Epoch is demonstrated in figure 7 while its Loss vs Epoch analysis is represented by Figure 8.

6.0 Conclusion

This study used Random Forest (RF) and Deep Convolutional Neural Networks (DCNN) to offer a thorough analysis on the creation of a unified model for the detection of fraudulent profiles and stalking prediction. The study has shown that the machine learning algorithms RF and DCNN may be used to identify bogus profiles and anticipate

stalking on social media sites, offering a useful tool for social media profile upkeep and administration. According to the study, a combined framework that makes use of RF and DCNN outperforms other techniques in terms of computational efficiency and accuracy. Because of the algorithms' capacity for learning and adaptation, they are especially well-suited for real-time applications, in which the model must be updated in real-time as new data is generated. The study has also demonstrated how crucial feature selection and preprocessing are to raising the model's efficiency. It has been demonstrated that using the right characteristics and preprocessing them with care can significantly increase the detection and prediction accuracy.

Although the results are encouraging, it is crucial to remember that a number of variables, The RF&DCNN model's performance can be influenced by various factors, such as the volume and caliber of training data, the choice of hyperparameters, and the intricacy of the social media network. Therefore, future efforts should concentrate on further refining these elements to raise the model's resilience and dependability. Furthermore, the finding has created fresh directions for further investigation. For example, combining RF and DCNN with additional machine learning methods may result in prediction models that are even more precise and effective. It might also be investigated to apply the concept to other similar issues including identifying fake news, stopping the construction of false accounts, and stopping online attacks and data exploitation. In conclusion, by presenting and validating the application of RF and DCNN, this dissertation has significantly advanced the field of anomaly detection and prediction on social media platforms. The research's conclusions not only offer a fresh viewpoint on the issue, but they also present workable answers that the business may quickly adopt.

6.1 Future work

- Further research could look into creating a model that stops the creation of phony profiles as a viable future direction. Establishing a model that stops the formation of fraudulent profiles is just as vital as detecting phony accounts when they are present in a network.
- The identification and categorization of bogus news could be a viable future avenue.

References

Asante, A. & Feng, X. (2021). Content-based technical solution for cyberstalking detection. *3rd International Conference on Computer Communication and the Internet (ICCCI).*

Balakrishnan, V., Khan, S. & Arabnia, H.R. (2020). Improving cyberbullying detection using Twitter users' psychological features and machine learning. *Computer & Security, 90*(3), 101710.

Beatriche, G. (2018). Detection of fake profiles in online social networks. *Proceedings of the International Conference on Advances in Computing and Communication Engineering (ICACCE), Paris, France.*

Bhosale, R. & Mane, V. (2024). Enhancing user trust: A novel hybrid model to detect fake profiles in online social networks. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(13s), 542.

Chakraborty, P., Shazan, M., Nahid, M., Ahmed, M. & Talukder, P. (2022). Fake profile detection using machine learning techniques. *Journal of Computer and Communications, 10*(10), 74-87.

Eastee, V.D.W. & Jan, E. (2018). Using machine learning to detect fake identities: Bots vs Humans. Retrieved from https://www.researchgate.net/publication/322650456_Using_Machine_Learning_to_Detect_Fake_Identities_Bots_vs_Humans

Egele, M., Stringhini, G., Stringhini, G. & Vigna, G. (2015). Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing 14(99)*. Retrieved from https://www.researchgate.net/publication/281768604_Towards_Detecting_Compromised_Accounts_on_Social_Networks

El-Azab, A., Idrees, A.M., Mahmoud, M. & Hefny, D.H. (2016). Fake account detection in Twitter based on minimum weighted feature set. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *10*(1), 13–18.

Fatih, C. A. & Esat, M.K. (2019). Identification of spurious and automated records that lead to a phony Instagram joint effort. *Social Network Analysis and Mining*, *4*(1). Retrieved from https://doi.org/10.1007/s13278-014-0194-4

Gayatri, N., Vaibhav, D., Kajal, D., Shraddha, G., Kulkarni, P.R. (2020). Detection of fake twitter accounts with machine learning algorithms. Retrieved from https://ijirt.org/master/publishedpaper/IJIRT150525_PAPER.pdf

Harish, K., Naveen, R., Kumar, J., & Briso, B. (2023). Fake profile detection using machine learning. *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 10*(2), 719-725. Retrieved fromhttps://doi.org/10.32628/IJSRSET2310264.

Huang, C., Fang, Y., Yang, S., & Zhao, B. (2021). Cyberbullying detection in social networks using Bi-gru with self-attention mechanism. *Information, 12*(4), 171.

Mohammed, A. (2020). Early detection of similar fake accounts on twitter using the random forest algorithm. *International Journal of Advanced Research in Engineering and Technology (IJARET), 11*(12), 611-620. Retrieved from http://iaeme.com/Home/issue/IJARET?Volume=11&Issue=12

Prathyusha, T., Sai-Kumar, T.N., Vishnu, E.P., & Vijaykanth, T.R. (2021). Fake account detection using machine learning. *International Journal of Creative Research Thoughts (IJCRT), 9*(6), e804-e807. Retrieved from https://ijcrt.org/papers/IJCRT2106559.pdf

Ratkiewicz, J., Conover, M. D., Meiss, M., Gonc, B., Flammini, A., & Menczer, F. (2010). Detecting and tracking political abuse in social media. Retrieved from https://ojs.aaai.org/index.php/ICWSM/article/view/14127

Reza, R. R., & Soheila, K. (2020). Detecting fake accounts on Twitter social network using multi-objective hybrid feature selection approach. *Webology, 17*(1), 1-18.

Saberi, A., Vahidi, M., & Bidgoli, B.M. (2007). *Learn to detect phishing scams using learning and ensemble methods*. *IEEE*, 311–314. Retreived from https://www.researchgate.net/publication/4310062_Learn_to_Detect_Phishing_Scams_Using_Learning_and_Ensemble_Methods

Saeid, S. (2020). An efficient method for detection of fake accounts on the instagram platform. *Revue d'Intelligence Artificielle, 34*(4), 429-436.

Secchiero, M. (2012). FakeBook: Detecting fake profiles in on-line social networks. *IEEE*. Retrieved from DOI: 10.1109/ASONAM.2012.185

Yazan, J. S. (2015). Thwarting fake OSN accounts by predicting their victims. Retrieved from https://dl.acm.org/doi/10.1145/2808769.2808772