Grenze International Journal of Engineering and Technology, June Issue

Identification of Phony Profiles on Social Platforms using Random Forest and Decision Tree Algorithms

M.S.P. Durga Rao[1], U Naga Nandini[2], T Nandhini[3], R Navya Sree[4] and V Lahari[5]
[1-5]Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India
Email: mspdurgarao1244@gmail.com, udayagirinaganandini@gmail.com, theerthagirinandhini@gmail.com, ratakondanavya03@gmail.com, laharivangimalla@gmail.com

***Abstract*— Given the pervasiveness of social media in today's world, the problem of identifying fraudulent accounts on sites like Social Media has taken on great significance. Python is the main technology used in this project, "Social Media Fake Account Detection using Machine Learning," to address this issue. The system employs 2 effective machine learning algorithms, the Classifier of Random Forests and its Decision Tree Classifier, to do this. The random forest classification system performs exceptionally well, achieving an accuracy of 100% on the training database and a startling accuracy rate of 93% on the test dataset. This Decision Tree Classifier simultaneously shows its efficiency by achieving ninety-two testing and ninety-two accuracy when trained. 576 records make up the dataset used for this project, and each record has 12 unique features. The presence or absence of a profile picture, the ratio of numbers of individuals in usernames and passwords the splitting of the full name into phrase tokens, the equality of login credentials and full names, the total length of user bios, which stands for the existence of other people's URLs, the private position of user accounts, the number of posts, the number of followers, the amount of takes into account followed, the and the last classification of a profile as "phony" or "Real" are just a few of the important aspects of online social networking user profiles covered by these characteristics. This project seeks to provide an efficient and dependable approach for recognizing Fake user profiles using Python and these advanced machine learning models. It contributes to keeping the software's validity and privacy of users intact.**

***Index Terms*— Python, Decision Tree Algorithm, Random Forest Algorith**

I. INTRODUCTION

The application of machine and deep learning paradigms has been prolific across a wide spectrum of academic fields, effectively spanning disciplines that include agronomy [1], spectral analysis [2], [3], cellular biology [4], and healthcare [5], [6].The act of inventing an online identity to mislead or hide one's genuine identity is known as creating a false account on social media. This behavior is frequently coupled to different goals, including disseminating false information, cyberbullying, anonymity, and stalking[7][8]. To create an online persona that appears authentic, the procedure usually involves fabricating names, images, and biographical data[9]. To participate in activities, they would not link with their own identity, people may turn to making up fake accounts[10][11]. This might be anything from publishing contentious images to falling for online fraud or scams. Social media companies spend money on moderation tools and algorithms to identify and block these phony accounts, which endanger user security and the platform's integrity[12].

Grenze ID: 01.GIJET.10.2.837

Furthermore, using false accounts compromises the legitimacy and trust of online interactions, which brings up ethical issues[13]. To keep the internet safe and reliable, users must stay alert and report any questionable accounts. To prevent the establishment and spread of false accounts, platforms are always improving their security protocols, which highlights the significance of practicing responsible digital citizenship.

Without explicit programming, a collection of computer algorithms referred to as "machine learning" can learn from examples and improve itself. Machine learning is a subset of artificial intelligence[14][15]. It forecasts output by using data and mathematical methods, which can be used to get insightful knowledge. The breakthrough is the notion that a machine can autonomously gain insight into information (i.e., an example) and produce precise results. Machine learning is closely linked to Statistical prediction and data mining. After ingesting input, the system applies an algorithm to produce replies[16][17][18].

The brain that does all the learning is machine learning. Machine learning is comparable to human learning. Experience teaches humans new things[19]. Predicting becomes easier with the most information we have. In comparison, we're given less likelihood to succeed under an unfamiliar circumstance compared to an existing context[20][21]. The machine shows an explanation to produce an accurate prediction. The machine can determine the result when we offer it another similar example. But much like a human, the machine finds it difficult to forecast if it is fed an example that hasn't been seen before[22][23]. Learning and inference are machine learning's main goals. Initially, the machine gains knowledge by identifying patterns. The data is what led to this discovery. Selecting the right data to feed the computer is an essential skill for a data scientist.

A collection of characteristics utilized to address a problem is called a characteristics array. Applying an aspect of information to a problem can be thought of as an attribute array[24].

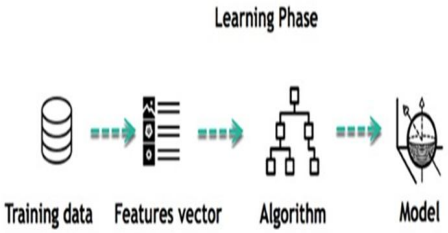

Figure (i). Learning Phase

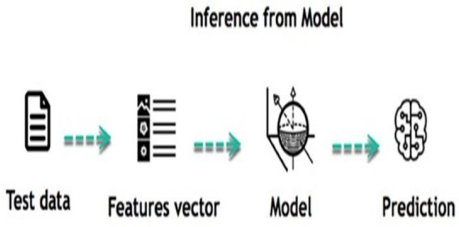

Figure (ii). Inference Model

A. Random Forest Algorithm

A flexible and potent machine learning approach for classification, as well as regression applications, is called Random Forest. It falls within the group of ensemble learning, which aims to increase overall reliability and precision by mixing predictions from several models[25]. The decision trees make the basis for the "forest" in the Random Forest model, and the introduction of randomization in the two training and prediction stages is referred to as the "random"part.

By choosing portions of the training phase's data and characteristics at random, Random Forest creates several decision trees throughout the training phase. This procedure called bagging (Bootstrap Aggregating), improves the model's capacity for generalization while assisting in the reduction of overfitting[26].

This algorithm's excellent precision and capacity to manage complicated data sets have made it popular in a variety of fields, such as the detection of images, finance, and healthcare. Random Forest is a widely used option in the machine learning world due to its adaptability, resilience, and capacity to handle both numerical and categorical input[27]. To get the best throughput for a given task, you must carefully adjust its hyperparameters.

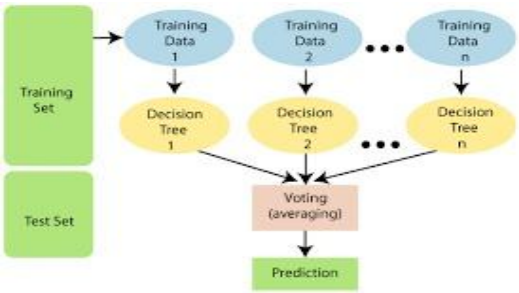

Figure (iii). Random Forest Flowchart

B. Decision Tree Algorithm

A Decision Tree is an important machine-learning approach for regression and classification work that has a tree-like structure. Internal nodes reflect feature-based decisions, locations reflect results, and leaf nodes show final predictions[26]. The primary goal is to produce identical subgroups based on the target parameter. During the process of making decisions, the dataset is iteratively partitioned based on features that provide the greatest information gain or reduce Maximum impurity. Gini impurity assesses the disorder in data points, whereas gain of information evaluates how successfully a characteristic distinguishes values.

Model decisions can be better explained by using decision trees because they are easily interpreted and visually simple. They can, however, overfit and end up collecting disturbance in training data. Tree depth restrictions or trimming can be used to remedy this. Decision trees might not apply well and are susceptible to fluctuations inthedata[27].

Decision trees are used in ensemble approaches such as Random Forests to create strong models. Because decision trees are a natural way to extract insights from large, complicated information, they are used in marketing, finance, and healthcare. Applying decision trees in machine learning models requires carefully a tuned hyperparameter and correcting excessive fitting.

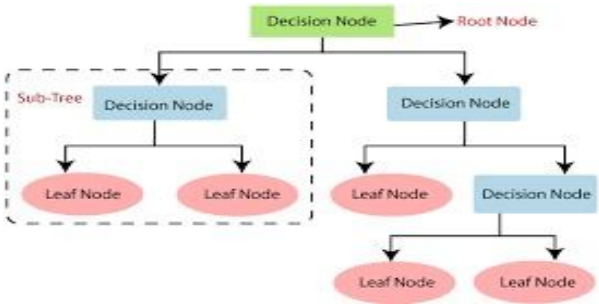

Figure (iv). Decision Tree Flowchart

II. METHODOLOGY

***Dataset*:** The dataset consists of 576 individual data. There are 12 columns in the dataset.

Data Collection and Preprocessing: Gathering information from social media platforms, such as activity logs, interaction patterns, and profile features, is the initial stage. To guarantee the consistency and quality of the dataset, preprocessing methods such as feature scaling, data cleaning, and missing value imputation are used. To reduce dimensionality and emphasize informative qualities, feature selection techniques can also be used.

Feature Engineering: To improve model performance, feature engineering is essential for obtaining valuable information from unprocessed data. The behavioral and structural traits of both real and fake accounts are captured by features including interaction variety, posting frequency, profile completeness, and language patterns. The selection and development of pertinent features are guided by domain expertise and exploratory data analysis.

Model Training: The constructed feature set is applied to learn the Random Forest and Decision Tree algorithms to identify trends that point to fraudulent profiles. Using bootstrapped samples of the data, several decision trees are constructed using the Random Forest group learning approach, which then aggregates the trees' predictions to improve resilience and generalization. Decision trees, on the other hand, create a hierarchical structure for categorization by recursively partitioning the feature space according to attribute splits.

Evaluation and Validation: The trained models' performance is assessed by the use of suitable measures, including F1-score, accuracy, precision, and recall. Techniques for cross-validation, such as k-fold cross-validation, guarantee objective evaluation of the model's performance over several data subsets. In addition, the models' efficacy in identifying fraudulent profiles is evaluated by real-world testing or validation on a holdout dataset.

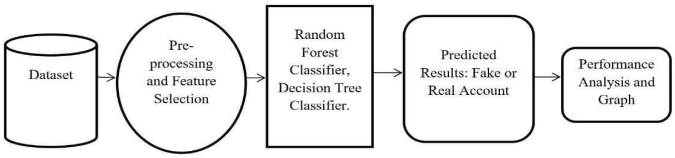

Figure (v). System Model

DATA FLOW DIAGRAM

An input dataset is first used in the process, and preprocessing is applied to improve its quality and suitability for further analysis. This includes operations like encoding categorical variables, scaling features, and handling missing values. The dataset is preprocessed and then divided into two sets: a training set that is used to construct the model and an evaluation set that is used to assess the model's performance.

On the chosen dataset, two classification algorithms—the Random Forest Classifier and the Decision Tree Classifier—are trained. While the Decision Tree Classifier constructs a tree structure for prediction, the Random Forest Classifier develops an ensemble of decision trees, offering robustness and minimizing overfitting. The testing dataset is subsequently subjected to the trained model's application for classification and prediction. The goal here is to distinguish between authentic and fraudulent accounts. The model classifies the testing data by applying its learning patterns and decision-making abilities. Ultimately, the anticipated outcomes are produced, showing if an account is considered authentic or fraudulent. The useful metric for evaluating the model's ability to differentiate within the two classes is this predicted result. For the specific purpose of distinguishing between phony and real accounts, this organized approach encompasses the phases of data preliminary processing, model training, hyperparameter optimization, and predictive analysis.

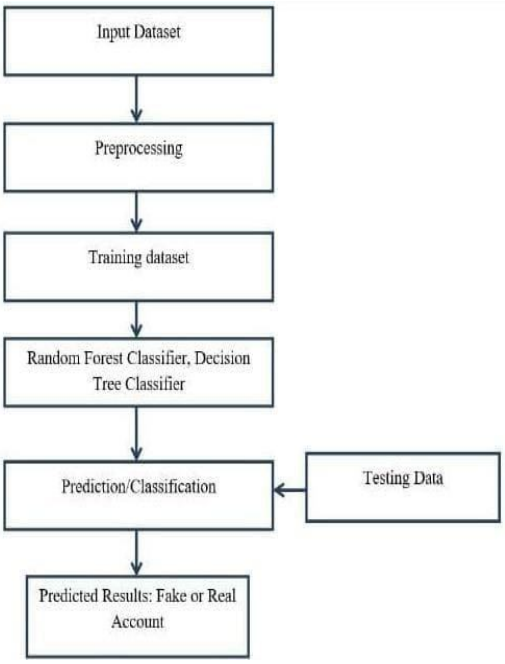

Figure (vi). Data Flowchart

III. SYSTEM ANALYSIS

The suggested Social Media fake account identification solution is built on a solid foundation of Python, a flexible and popular programming language in the data analysis and machine learning domains. The Random Forest Classification and the Decision Tree Classification are two important machine learning models that the system uses to improve its ability to discriminate between real and phony Social Media accounts. Python is used to create the system, and it provides a rich ecosystem of modules and tools for modeling, assessment, and preparing data. Python is a great fit for this project because of its many machine-learning tools and versatility. The suggested solution uses the Random Forest Classifier and Decision Tree Classification technique machine learning algorithms to assess Social Media profiles for authenticity as a whole. The Random Forest Classifier model demonstrates its ability to generalize well and generate correct predictions by achieving a solid 93% precision score on the test dataset and a spectacular 100% precision score on the training dataset. The Decision Tree Classifier has a 92% training precision score and a 92% test accuracy, indicating its continued applicability for the fraudulent account detection problem. The 576 entries in the dataset that the system uses to run its operations are enhanced with 12 distinct attributes that cover a range of aspects related to Social Media accounts. Among these features are the following: External URL, Name==username, Display picture, Number of digits in the username,

Fullname words, and Number of digits in the full name. personal, Fake, #Posts, #Followers, #Follows. The suggested approach addresses certain drawbacks and expands on the advantages of the current system, which attained remarkable accuracy levels. To predict thorough and accurate results for Social Media fake account detection, it combines a variety of algorithms, strong feature engineering, interpretability, adaptation to new threats, and increased efficiency. The main goal of this technology is to make the Internet community platform more reliable and secure.

IV. Result Analysis

System testing, a crucial stage in the software development life cycle, evaluates a software system's overall functionality, quality, and performance. It is an extensive and methodical procedure designed to find errors, make sure the system satisfies requirements, and confirm that it is ready for use. Delivering dependable, sturdy, and superior software solutions depends heavily on system testing.

RandomForestClassifier

	Recall	Precision	F1-score
0	0.90	0.98	0.94
1	0.98	0.90	0.94

Figure (vii). Random Forest Classifier

For a binary classification problem, where there were two unique classes labeled 0 and 1, a Random Forest Classifier was used. For both classes, the relevant performance metrics—Recall, Precision, and F1-score—were calculated. With a high recall (sensitivity) of 0.90 for Class 0, the classifier proved to be highly effective in properly identifying instances of this class out of all the relevant examples. In a notable higher accuracy of 0.98 for Class 0, there was little chance of false positives. The F1-score for Class Zero, which evaluates how well Precision and Recall are balanced, was determined to be 0.94. This indicates a strong performance in identifying genuine positive cases and reducing false positives. On the other hand, the classifier demonstrated a high Precision of 0.98 for Class 1, suggesting a low false positive rate. Class 1's recall, on the other hand, was marginally lower at 0.90, indicating that the classifier may have missed a small number of instances from this class. The F1-score for Class 1 One was 0.94, It was likewise the same as Class 0's.

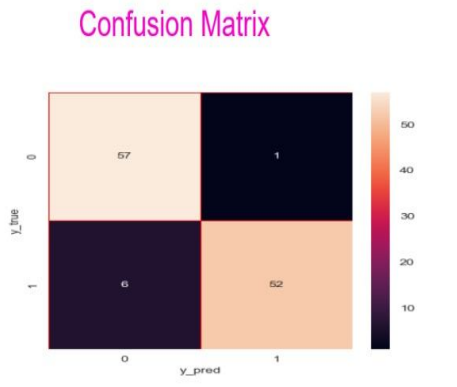

Figure (viii). Confusion matrix of Random Forest

Based on the F1 scores, the Random Forest Classifier performed admirably overall in both classes, striking better stability in the middle of precision and recall. In situations when it's critical to strike stability in the middle of false positives and false negatives, the classifier's resilience in managing both positive and negative occurrences is essential. These results highlight the Random Forest model's effectiveness in the particular categorization task that was studied. Furthermore, the thorough examination of each statistic offers insights into the classifier's advantages as well as potential places for improvement for particular class considerations. These findings provide important information for the discussion and conclusion portions of a journal publication on the classification problem at

hand, which is crucial for practitioners and decision-makers who depend on precise predictions in the researched area.

DecisionTreeClassifier

	Recall	Precision	F1-score
0	0.89	0.97	0.93
1	0.96	0.88	0.92

Figure (ix). Decision Tree Classifier

For the binary classification challenge, where the classes were labeled as 0 and 1, the Decision Tree Classifier performed differently than the Random Forest Classifier. For both courses, the primary performance metrics—Recall, Precision, and F1-score—were calculated. The Decision Tree Classifier's Recall for Class 0 was 0.89, demonstrating its ability to accurately identify instances of this class out of all the relevant examples. With a better accuracy of 0.97 for Class Zero, there was little chance of false positives. Class Zero's F1-score of 0.93 indicates a balanced performance in eliminating false positives and capturing true positive situations.The Decision Tree Classifier denotes a peak recall of 0.96 for Class 1, demonstrating its efficacy in detecting positive cases. In addition to, Class 1's Precision was somewhat lower at 0.88, indicating a considerably higher rate of false positives. Class 1 has an F1 score of 0.92, which shows that Precision and Recall are well-balanced. When comparing the Decision Tree Classifier's performance to the Random Forest Classifier that was previously mentioned, the Decision Tree model produced generally somewhat lower Precision and F1-score for both classes. It performed well in Recall, demonstrating its capacity to identify favorable examples, but there was a greater trade-off with Precision.

Figure (x). Confusion Matrix of Decision Tree

In summary, the Decision Tree Classifier outperforms other methods for the binary classification job, particularly when it comes to recall. But for a thorough analysis, Precision and F1 scores must be taken into account. Decision trees could be a good option when a high recall rate is desired, but precision must be carefully balanced, particularly if reducing false positives is a crucial demand. These results offer insightful information for decision-makers and industry practitioners, broadening our understanding of the applicability of Decision Tree models for particular classification tasks.

V. Conclusion

To sum up, the project "Identification of Phony Profiles On Social Platforms Using Random Forest and Decision Tree Algorithms" offers a thorough and practical way to deal with the problem of distinguishing real Social Media accounts from fraudulent ones. This system, which was created with Python and makes use of the Random Forest Classifier and the Decision Tree Classifier, two potent machine learning models, has been demonstrated to be highly efficient and dependable in its operation.The 576 entries in the dataset that the system uses to work are enhanced with 12 unique attributes that represent different views of Social Media accounts, including the existence of account images, the format of profile names and complete names, the length of the bio, external URLs, and more. The system can identify phony accounts with accuracy and consistency because of these capabilities and strong feature engineering.

Improvements in content analysis, interpretability, adaptation to new threats, and privacy concerns all add to the system's effectiveness and user confidence. Using a variety of classifiers and ensuring algorithm diversity allows

for a more thorough assessment of Social media profiles. The Random Forest Classifier's test accuracy stands at 93%, while the Decision Tree Classifier's test accuracy is at 92%. These results demonstrate the proposed system's great capacity to generalize and reduce false negatives and positives. These characteristics are necessary to keep the Social platform safe and secure.As a result, the project solves the shortcomings of the current system in addition to enhancing its strengths. It provides a comprehensive solution that attempts to better Instagram's security, reliability, and user experience by efficiently detecting and reducing the existence of phony accounts.

REFERENCES

[1] R. Mohamed Yousuff and M. Rajasekhara Babu, "Improving the Accuracy of Prediction of Plant Diseases Using Dimensionality Reduction-Based Ensemble Models," in Emerging Research in Data Engineering Systems and Computer Communications, 2020, pp. 121–129, doi: 10.1007/978-981-15-0135-7_11.

[2] M. Yousuff and R. Babu, "Enhancing the classification metrics of spectroscopy spectrums using neural network based low dimensional space," Earth Sci. Informatics, vol. 16, no. 1, pp. 825–844, 2023, doi: 10.1007/s12145-022-00917-1.

[3] M. Yousuff and R. Babu, "Deep autoencoder based hybrid dimensionality reduction approach for classification of SERS for melanoma cancer diagnostics," J. Intell. Fuzzy Syst., vol. 43, no. 6, pp. 7647–7661, 2022, doi: 10.3233/JIFS-212777.

[4] M. Yousuff, R. Babu, and A. Rathinam, "Nonlinear dimensionality reduction based visualization of single-cell RNA sequencing data," J. Anal. Sci. Technol., vol. 15, no. 1, p. 1, 2024, doi: 10.1186/s40543-023-00414-0.

[5] A. R. Mohamed Yousuff, R. Anusha, J. Vijayashree, and J. Jayashree, "The Role of Artificial Intelligence for Intelligent Mobile Apps," in Emerging Technologies for Sustainable and Smart Energy, 1st ed., A. Sircar, G. Tripathi, N. Bist, K. A. Shakil, and M. Sathiyanarayanan, Eds. CRC Press, 2022.

[6] R. Mohamed Yousuff, M. Zainulabedin Hasan, R. Anand, and M. Rajasekhara Babu, "Leveraging deep learning models for continuous glucose monitoring and prediction in diabetes management: towards enhanced blood sugar control," Int. J. Syst. Assur. Eng. Manag., Jan. 2024, doi: 10.1007/s13198-023-02200-y.

[7] E. Karunakar, V. D. R. Pavani, T. N. I. Priya, M. V. Sri, and K. Tiruvalluru, "Ensemble fake profile detection using machine learning (ML)," J. Inf. Comput. Sci., vol. 10, pp. 1071–1077, 2020.

[8] P. Wanda and H. J. Jie, "Deep profile: utilizing dynamic search to identify phony profiles in online social networks CNN" J. Inf. Secure. Appl., vol. 52, pp. 1–13, 2020.

[9] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS spam," Future Gener. Comput. Syst., vol. 102, pp. 524–533, 2020.

[10] R. Kaur, S. Singh, and H. Kumar, "A modern overview of several countermeasures for the rise of spam and compromised accounts in online social networks," J. Netw. Comput. Appl., vol. 112, pp. 53–88, 2018.

[11] G. Suarez-Tangil, M. Edwards, C. Peersman, G. Stringhini, A. Rashid, and M. Whitty, "Automatically dismantling online dating fraud," IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 1128–1137, 2020.

[12] M. U. S. Khan, M. Ali, A. Abbas, S. U. Khan, and A. Y. Zomaya, "Segregating spammers and unsolicited bloggers from genuine experts on Twitter," IEEE Trans. Dependable Secure Comput., vol. 15, no. 4, pp. 551–560, Jul./Aug. 2018.

[13] V. Balakrishnan, S. Khan, and H. R. Arabnia, "Improving cyberbullying detection using Twitter users' psychological features and machine learning," Comput. Secur., vol. 90, 2020, Art. no. 101710.

[14] Georgios Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos, "Detecting social network profile cloning," 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Seattle, WA, USA, 2011, pp. 295-300, doi: 10.1109/PERCOMW.2011.5766886.

[15] Monther Aldwairi, and Ali Alwahedi, "Detecting Fake News in Social Media Networks", Procedia Computer Science, Volume 141, 2018, Pages 215-222; https://doi.org/10.1016/j.procs.2018.10.171

[16] Buket Erşahin, Özlem Aktaş, D. Kılınç, and C. Akyol, "Twitter fake account detection," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 2017, pp. 388-392, doi: 10.1109/UBMK.2017.8093420.

[17] Kumud Patel, Saijshree Srivastava, and Sudhanshu Agrahari, "Survey on Fake Profile Detection on Social Sites by Using Machine Learning Algorithm," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 1236-1240, doi: 10.1109/ICRITO48877.2020.9197935.

[18] Alexey D.Frunze and Aleksey A. Frolov, "Methods for Detecting Fake Accounts on the Social Network VK," 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia, 2021, pp. 342-346, doi: 10.1109/ElConRus51938.2021.9396670.

[19] M. BalaAnand, S. Sankari, R. Sowmapriya, and S. Sivaranjani, "Recognizing fraudulent users on social networks through their nonverbal cues," Int. J. Technol. Eng. Syst., vol. 7, no. 2, pp. 157–161, 2015.

[20] A. M. Meligy, H. M. Ibrahim, and M. F. Torky, "Identifier checker tool for online social networks to identify false profiles," Int. J. Comput. Netw. Inf. Secure., vol. 9, no. 1, pp. 31–39, 2017.

[21] M. Fire, A. Elyashar, and Y. Elovici, "Friend or enemy? identification of fake profiles in internet Social networks", Social Netw. Anal. Mining, vol. 4, no. 1, 2014, Art. no. 194.

[22] Egele, G. Stringhini, C. Kruegel, and G. Vigna, "IEEE Trans. Dependable Secure Comput., "For detecting compromised accounts on social networks," IEEE Trans. Dependable Secure Comput., vol. 14, no. 4, pp. 447–460, Jul./Aug. 2017.

[23] K. Chakraborty, S. Bhattacharyya, and R. Bag, “A survey of sentiment analysis from social media data,” IEEE Trans. Comput. Social Syst., vol. 7, no. 2, pp. 450–464, Apr. 2020.
[24] S. Lee and J. Kim, “WarningBird: IEEE Trans. Dependable Secure Comput., "A near real-time detection method for suspicious URLs in Twitter stream,” IEEE Trans. Dependable Secure Comput., vol. 10, no. 3, pp. 183–195, May/Jun. 2013.
[25] H. Drucker, D. Wu, and V. N. Vapnik, “To classify spam, support vector machines,” IEEE Trans. Neural Net., vol. 10, no. 5, pp. 1048–1054, Sep. 1999.
[26] Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min, “Real-time drifted Twitter spam detection using statistical features,” IEEE Trans. Inf. Forensics Secur., vol. 12, no. 4, pp. 914–925, Apr. 2016.
[27] Chen et al., “Streaming spam tweets detection using machine learning: performance evaluation,” IEEE Trans. Comput. Social Syst., vol. 2, no. 3, pp. 65–76, Sep. 2015.