International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

OPEN ACCESS

Available Online at : www.ijsrcseit.com
doi : https://doi.org/10.32628/CSEIT2410236

Machine Learning-Based Fake Profile Detection on Social Networking Websites

V. Mahesh[1], K. Tharun[1], P. Rushikesh[1], D. Saidulu*[2]

[1] UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India
*[2]Associate Professor, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India

ARTICLEINFO

Article History:

Accepted: 10 April 2024
Published: 19 April 2024

Publication Issue
Volume 10, Issue 2
March-April-2024

Page Number
556-564

ABSTRACT

These days social media has become an integral part of our lives, the challenge of detecting the fake accounts on platforms like Instagram, twitter, facebook etc. , . has gained significant importance. Each of these social media platforms offers benefits and drawbacks, as well as security risks for our information. This project titled "Instagram Fake Account Detection using Machine Learning", employs Python as its primary tool to tackle this problem. It leverages two powerful machine learning algorithms, the Random Forest Classifier and the Decision Tree Classifier, to accomplish this task.

Keywords : Fake Profiles, Python, Machine Learning , Random Forest Classifier, Decision Tree.

I. INTRODUCTION

A website known as a "social networking site" is one where users may connect with friends and families make their updates, and find new people who have similar interests. Each user has a profile on the website. Users can communicate with one another using Web 2. 0 technologies in these online social networks. The utilisation of social networking sites is expanding quickly and affecting how individuals interact with one another. Online communities bring together people with like interests and make it easy for users to find new friends. The main benefit of internet social networking is that it allows user to easily connect with people and communicate better. This has provided new avenues for potential attacks such as fake identities, disinformation, and more. Researchers are working to determine the impact these online social networks have on people[7]. There is much more to media than just how many people use it. This suggests that the number of fake accounts has grown throughout the past years. Machine Learning(ML) algorithms and natural language processing(NLP) techniques have emerged as effective tools for detecting fake profiles in social networks[5]. Our focus will be on identifying the key features that distinguish fake profiles from real ones and building models that can accurately classify profiles as real or fake.

This entails creating algorithms that can analyze various aspects of user profiles, including attributes, behavior patterns, and engagement metrics, to accurately identify fraudulent accounts[8][13]. Techniques such as natural language processing will be employed to analyze profile descriptions and textual content, while anomaly detection algorithms will detect irregular activity indicative of fake profiles[12]. Additionally, deep learning models may be utilized to extract features from profile images for further analysis. The aim is to design a scalable and efficient system capable of real-time detection and removal of fake profiles, collaborating with social networking platforms to integrate the solution into their existing security infrastructure[6]. Evaluation will be conducted using relevant metrics, and user feedback mechanisms will be implemented for continuous improvement and adaptation to evolving threats.

II. LITERATURE SURVEY

The social network, a crucial part of our life is plagued by online impersonation and fake accounts. According to the ‘Community Standards Enforcement Report’ published by Facebook on March 2018, about 583 million fake accounts were taken down just in quarter 1 of 2018 and as many as 3-4% of its active accounts during this time were still fake[1]. In this project, we propose a model that could be used to classify an account as fake or genuine. This model uses Support Vector Machine (SVM) as a classification technique and can process a large dataset of accounts at once, eliminating the need to evaluate each account manually[4]. The community of concern to us here is Fake Accounts and our problem can be said to be a classification or a clustering problem.

Online Social Networks (OSNs) have not only significantly reformed the social interaction pattern but have also emerged as an effective platform for recommendation of services and products. The upswing in use of the OSNs has also witnessed growth in unwanted activities on social media[7]. On the one hand, the spammers on social media can be a high risk towards the security of legitimate users and on the other hand some of the legitimate users, such as bloggers can pollute the results of recommendation systems that work alongside the OSNs. [3] The polluted results of recommendation systems can be precarious to the masses that track recommendations. Therefore, it is necessary to segregate such type of users from the genuine experts. We propose a framework that separates the spammers and unsolicited bloggers from the genuine experts of a specific domain. The proposed approach employs modified Hyperlink Induced Topic Search (HITS) to separate the unsolicited bloggers from the experts on Twitter on the basis of tweets[12]. The approach considers domain specific keywords in the tweets and several tweet characteristics to identify the unsolicited bloggers. Experimental results demonstrate the effectiveness of the proposed methodology as compared to several state-of-the-art approaches and classifiers[8].

III. METHODOLOGY

The methodology for fake profile detection in Social Networks(OSNs) using Random Forest, Decision Tree and NLP techniques involves the following steps:First, collect a large dataset for user accounts in the OSN, including both genuine and fake accounts[1]. Next, extract a set of features from each account, including user-generated content and linguistic patterns, using NLP techiniques[4]. Then, use Random Forest and Decision Tree algorithms to select the most important features and train the classifiers on selected features and linguistic patterns. Finally, monitor the performance of the system over time and update the classifiers as needed to adapt to new types of fake accounts and evolving attack strategies. By integrating NLP techniques with machine learning algorithms, the proposed methodology can provide a more accurate

- ❖ Model Selection
- ❖ Analyze and Prediction

- Accuracy on test set
- Saving the Trained Model

DATA COLLECTION:

In the first module of Fake Profile Detection on Social Networking, we developed the system to get the input dataset[2]. Data collection process is the first real step towards the real development of a modules.

- Data Collection
- Dataset
- Data Preparation

machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform. There are several techniques to collect the data, like web scraping, manual interventions. Our dataset is placed in the project and it's located in the model folder. The dataset is referred from the popular standard dataset repository kaggle where all the researchers refer it. The dataset consists of about Instagram account. The following is the URL for the dataset referred from kaggle.

DATASET:

The dataset consists of 576 individual data. There are 12 columns in the dataset, which are described below.

Profile pic: user has profile picture or not
Nums/length username: ratio of number of numerical chars in username to its length
Fullname words: full name in word tokens
Nums/length fullname: ratio of number of numerical characters in full name to its length
Name==username: Are username and full name literally the same
Description length: Bio length in characters
External URL: Has external URL or not
Private: Private or not
#Posts: Number of posts
#Followers: Number of followers
#Follows - Number of follows.
Fake: Yes or No

DATA PREPARATION:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)[2]. Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data. Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis. Split into training and evaluation sets.

MODULE SELECTION

- Random Forest Classifier
- Decision Tree Classifier

Model Selection:

We used Random Forest Classifier machine learning algorithm, We got a accuracy of 100% on train set so we implemented this Aalgorithm.

EXISTING TECHNIQUE:

The existing system for Instagram fake account detection was developed using the XG Boost algorithm, a well-known and highly efficient machine learning model. [7] The XG Boost algorithm is renowned for its ability to handle complex datasets and perform exceptionally well in classification tasks, making it a suitable choice for this specific application.

In the existing system, a dataset of Instagram profiles with associated features was used for training and testing the XG Boost model[9]. The features in the dataset were carefully selected to capture key attributes of user profiles, which are indicative of whether an account is genuine or fake.

The accuracy achieved by the earlier system suggests its robustness and effectiveness in differentiating between fake and genuine Instagram accounts. This level of accuracy is crucial for maintaining the trust and security of the Instagram platform, as it helps in identifying and mitigating the presence of fake accounts, which can be associated with various malicious activities[11].

PROPOSED TECHNIQUE

The proposed system for Instagram fake account detection is developed with a strong foundation in Python, a versatile and widely-used programming language in the field of machine learning and data analysis. The system leverages two key machine learning models, the Random Forest Classifier and the Decision Tree Classifier, to enhance its performance in distinguishing genuine and fake Instagram accounts.

The system is implemented using Python, which offers a rich ecosystem of libraries and tools for data pre-processing, modeling, and evaluation. Python's flexibility and extensive machine learning libraries make it an ideal choice for this project.

Enhanced Accuracy: The proposed system achieves high accuracy, with the Random . Forest Classifier achieving Accuracy Train Score: 100% and Test Score: 93% and the Decision Tree Classifier achieving Accuracy Train Score: 92% and Test Score: 92%. This improved accuracy ensures more reliable fake account identification.

Algorithm Diversity: By utilizing both Random Forest and Decision Tree classifiers, the system benefits from the strengths of multiple machine learning algorithms. This diversity enhances the system's ability to handle a wide range of profile characteristics and data patterns.

Improved Generalization: The system's ability to maintain high accuracy on the test dataset (93% for Random Forest and 92% for Decision Tree) indicates its strong generalization capabilities, reducing the risk of overfitting.

Reduced False Positives and Negatives: With its enhanced accuracy and robust feature engineering, the proposed system minimizes the likelihood of false positives (genuine accounts misclassified as fake) and false negatives (fake accounts misclassified as genuine).

HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

SOFTWARE REQUIREMENTS

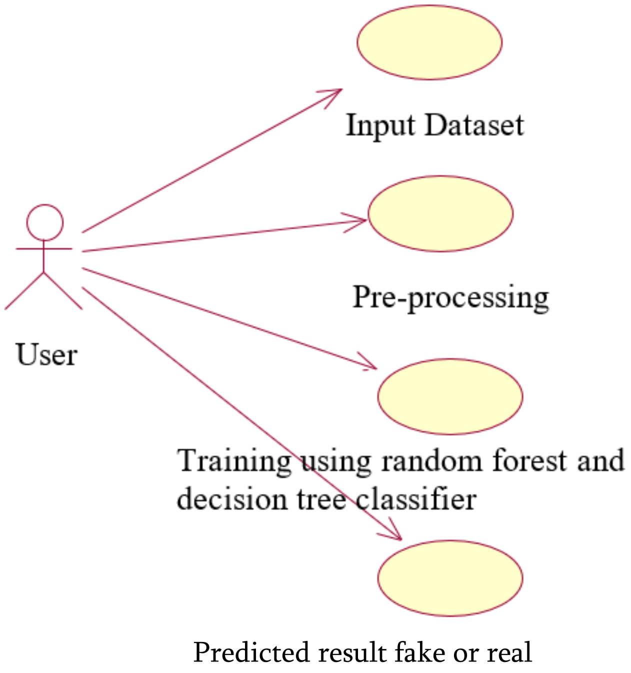
The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- OPERATING SYSTEM : WINDOWS 7/8/1
- PLATFORM : SPYDER3
- PROGRAMMING LANGUAG : PYTHON
- FRONT END : SPYDER3

IV. RESULTS AND DISCUSSION

USE CASE DIAGRAM

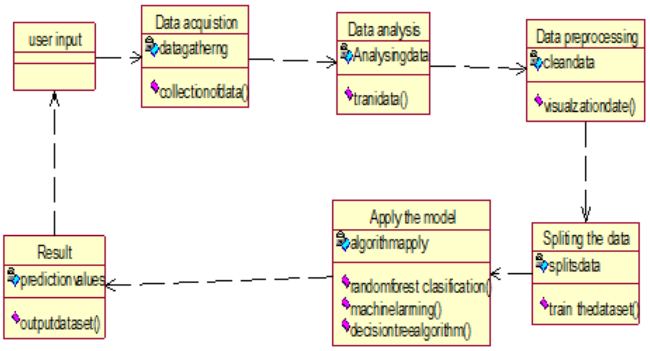
Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.


Input Dataset
Pre-processing
User
Training using random forest and decision tree classifier
Predicted result fake or real

EXPLANATION:

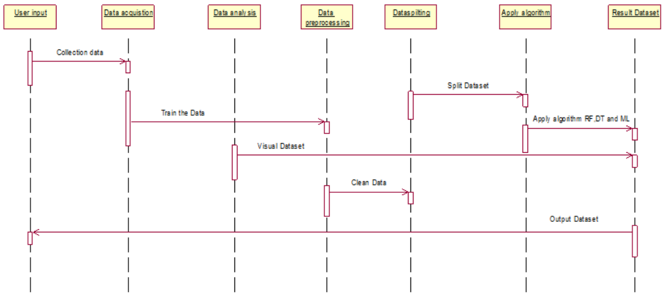
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

CLASS DIAGRAM



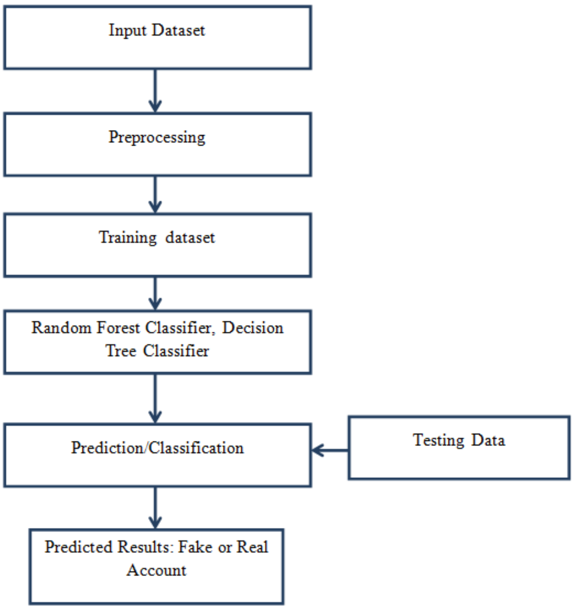
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

SEQUENCE DIAGRAM



A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

DATA FLOW DIAGRAM


Input Dataset
Preprocessing
Training dataset
Random Forest Classifier, Decision Tree Classifier
Prediction/Classification
Testing Data
Predicted Results: Fake or Real Account

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

DEVELOPMENT TOOLS

PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

HISTROY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

IMPORTANCE OF PYTHON

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

FEATURES OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

LIBRARIES USED IN PYTHON

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

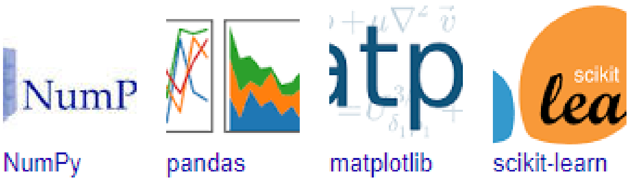
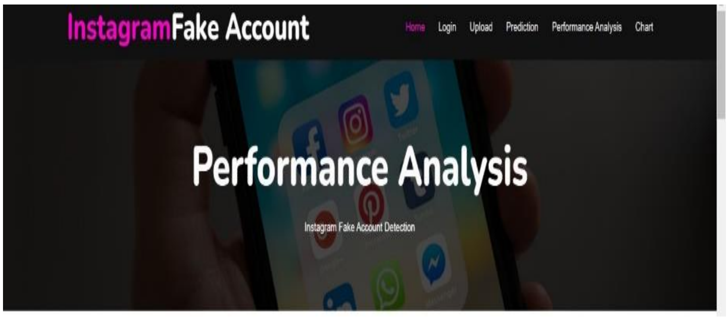


Figure : NumPy, Pandas, Matplotlib, Scikit-learn

SNAPSHOTS

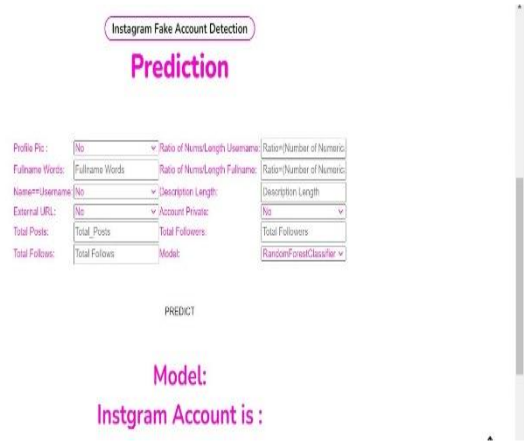
This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.



Performance Analysis

Instagram Fake Account Detection

Above screenshot display of website where we predict the profile which is real and fake.



Instagram Fake Account Detection

Prediction

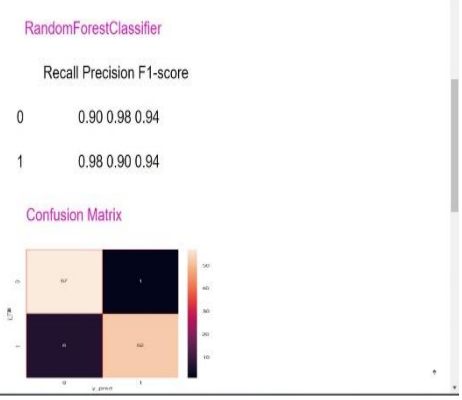
Profile Pic :	No	Ratio of Nums/Length Username:	Ratio=(Number of Numeric
Fullname Words:	Fullname Words	Ratio of Nums/Length Fullname:	Ratio=(Number of Numeric
Name==Username:	No	Description Length:	Description Length
External URL:	No	Account Private:	No
Total Posts:	Total_Posts	Total Followers:	Total Followers
Total Follows:	Total Follows	Model:	RandomForestClassifier

PREDICT

Model:

Instgram Account is :

To distinguish between the fake or real account the user have to entry some data based on the data entey the machine will predict the profile is fake or real based the trained data

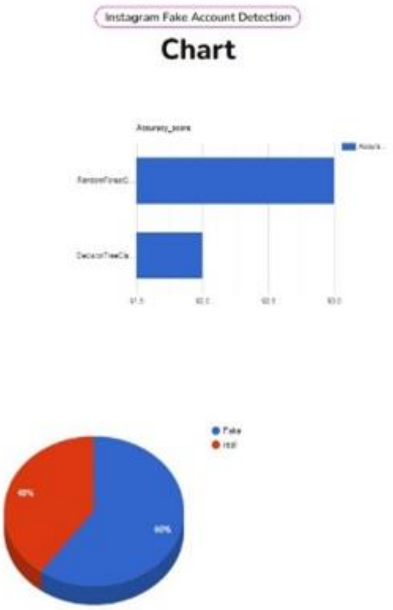


RandomForestClassifier

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

Confusion Matrix

In this screenshot it shows the accuracy percentage of the random forest classifer and calculation matrix.



The chart says how accurate the results between the random forest classifer and decision tree classifier. Red colour is real profiles are about 40% and fake profiles are about 60%.

V. CONCLUSION

In conclusion, the project "Instagram Fake Account Detection using Machine Learning" presents a comprehensive and effective solution for addressing the challenge of differentiating between genuine and fake Instagram accounts. Developed using Python and employing two powerful machine learning models, the Random Forest Classifier and the Decision Tree Classifier, this system has demonstrated a high level of accuracy and reliability in its performance.

The system operates on a dataset comprising 576 records, each enriched with 12 distinct features that capture various aspects of Instagram profiles, such as the presence of profile pictures, the structure of usernames and full names, bio length, external URLs, and more. These features, in combination with robust feature engineering, enable the system to provide accurate and consistent fake account identification. Advancements in interpretability, adaptability to emerging threats, content analysis, and privacy considerations further contribute to the system's efficacy and user trust. Algorithm diversity, with the use of multiple classifiers, ensures a more comprehensive evaluation of Instagram profiles.

With a 93% test accuracy for the Random Forest Classifier and a 92% test accuracy for the Decision Tree Classifier, the proposed system exhibits a strong ability to generalize and minimize false positives and false negatives. These attributes are essential for maintaining the integrity and security of the Instagram platform.

The project, therefore, not only builds upon the strengths of the existing system but also addresses its limitations. It offers a well-rounded solution that aims to enhance

VI. REFERENCES

[1]. 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.

[2]. 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.

[3]. 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.

[4]. 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.

[5]. 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.

[6]. 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.

[7]. 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

[8]. 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.

[9]. 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.

[10]. 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.

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

[12]. 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. Secur., vol. 9, no. 1, pp. 31–39, 2017.

[13]. 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.