ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/1301-1311
P.Ravali *et. al.*, / International Journal of Engineering & Science Research

Neural Network-Based Detection of Fraudulent Profiles In Social Media Platforms

Mr. K.Vijay Kumar[1], P.Ravali[2], P.Vyshnavi[3], P.Mithun[4], S.Nithin[5]

[1] Assistant professor, Department of CSE, Bhoj Reddy Engineering College for Women, India

[2,,43]B.Tech Students, Department of CSE, Bhoj Reddy Engineering College for Women, India

ABSTRACT

Social media platforms have become pervasive in modern society, offering opportunities for individuals to connect, share information, and engage in various activities. However, the rise of fraudulent activities, such as fake profiles, poses significant challenges to the integrity and security of these platforms. Traditional methods of detecting fraudulent profiles often rely on manual inspection or rule-based systems, which can be time-consuming and ineffective in identifying sophisticated fraudulent behavior. This study proposes a novel approach using neural networks for the automated detection of fraudulent profiles in social media platforms. By leveraging the power of deep learning techniques, the proposed system learns intricate patterns and features from large-scale datasets, enabling it to effectively distinguish between genuine and fraudulent profiles. The neural network model is trained on diverse sets of features, including user behavior patterns, content characteristics, network structure, and temporal dynamics, to capture the complex nature of fraudulent activities. Experimental results demonstrate the efficacy of the proposed approach in detecting fraudulent profiles with high accuracy and efficiency. Compared to traditional methods, the neural networkbased detection system achieves superior performance in terms of precision, recall, and F1-score. Moreover, the model exhibits robustness against various evasion techniques employed by fraudsters, making it suitable for real-world deployment in social media platforms. Social networking sites such as Facebook, Twitter, histogram, etc. are extremely famous among people. Users always interact with their friends via these social networks sites or media. They share their personal and public information using these social networks. an immense number of people use social networking sites due to their attractiveness. This fame causes problems to the websites due to the creation of fake accounts. The owners of fake accounts pull out personal information about other people and spread the fake data on social networks. In our proposed plan, we propose machine learning techniques such as Neural Networks and SVM for detecting fake accounts on Facebook or Twitter, or Twitter.

1. INTRODUCTION

In the rapidly evolving landscape of social media, the rise of fraudulent profiles poses a significant challenge to users, platforms, and cybersecurity experts alike. As the digital realm becomes increasingly intertwined with our daily lives, the need for robust mechanisms to identify and mitigate fraudulent activities is more pressing than ever before. In this context, the application of neural networks emerges as a beacon of hope, offering a sophisticated and dynamic solution to the complex problem of detecting fraudulent profiles.

Social media platforms serve as virtual arenas for billions of users worldwide to connect, share, and interact. However, this interconnectedness also presents a fertile ground for malicious actors seeking to exploit vulnerabilities for personal gain, be it

through identity theft, financial scams, or the dissemination of misinformation. Traditional methods of fraud detection often fall short in the face of rapidly evolving tactics employed by fraudsters, necessitating a paradigm shift towards more advanced and adaptive approaches.

Enter neural networks, a branch of artificial intelligence inspired by the complex interconnected structure of the human brain. These computational models excel at recognizing patterns, making them particularly well-suited for tasks such as image recognition, natural language processing, and, crucially, fraud detection. By leveraging vast amounts of data and sophisticated algorithms, neural networks can discern subtle indicators of fraudulent behavior that may elude human observers or conventional detection methods.The essence of neural network-based detection lies in its ability to learn and adapt in realtime, continuously refining its understanding of what constitutes legitimate user behavior versus suspicious activity. Through a process known as training, neural networks analyze vast datasets comprising examples of both genuine and fraudulent profiles, extracting underlying patterns and features that distinguish between the two.This process equips the neural network with the capability to generalize its learnings and accurately identify .

2-LITERATURE SURVEY

"Fraud Detection in Social Media: A Review" by Yang et al. (2019)

This comprehensive review examines various techniques and methodologies employed in fraud detection within social media platforms. The paper discusses traditional approaches, such as rule-based and statistical methods, as well as emerging techniques, including machine learning and neural network-based approaches. It provides insights into the challenges and opportunities associated with fraud detection in social media, laying the groundwork for further research in the field.

"Deep Learning for Fraud Detection: A Comprehensive Review" by Zhou et al. (2020)

Focusing specifically on deep learning techniques, this review paper provides a detailed overview of the application of neural networks in fraud detection across different domains, including finance, e-commerce, and social media. The authors explore various deep learning architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), highlighting their strengths and limitations in detecting fraudulent activities. The paper also discusses challenges related to data imbalances, interpretability, and scalability, offering valuable insights for researchers and practitioners.

Detecting Fake Accounts in Online Social Networks at the Time of Registrations" by Cao et al. (2019)

This research article proposes a neural network-based approach for detecting fake accounts on social media platforms at the time of registration. The authors leverage features extracted from user profiles and registration activities to train a deep neural network classifier. Through extensive experiments on real-world datasets, they demonstrate the effectiveness of their approach in accurately identifying fraudulent profiles, thereby mitigating the proliferation of fake accounts and improving the overall security of online social networks.

"Detecting Fake Accounts in Social Networks Using Deep Learning" by Sathyanarayana et al. (2018)

In this paper, the authors present a novel approach for detecting fake accounts in social networks using deep learning techniques. They propose a deep neural network architecture that combines

convolutional and recurrent layers to capture both spatial and temporal patterns in user behavior. By training the model on a large-scale dataset comprising genuine and fake accounts, they achieve high accuracy in distinguishing between the two, demonstrating the efficacy of their approach in combating fraudulent activities on social media platforms.

"A Survey of Fraud Detection Techniques in Social Media Networks" by Al-Tairi et al.(2020) This survey paper provides a comprehensive overview of fraud detection techniques specifically tailored to social media networks. It covers a wide range of methodologies, including traditional rule-based approaches, statistical techniques, machine learning algorithms, and deep learning models. The authors discuss the advantages and limitations of each approach, highlighting the role of neural networks in addressing the inherent challenges of detecting fraudulent activities in dynamic and heterogeneous social media environments.

B. D. Freeman et al. (2015) The identification of groups of fraudulent accounts, as opposed to a single fake account, was the primary emphasis of this work. This strategy resulted in the formation of a cluster, which was based on the characteristics that were supplied during the registration process, such as the registration date and the registration IP address. In order to train the model, Random Forest, Support Vector Machines (SVM), and Logistic Regression are used. SVM is utilized in order to determine whether or not the cluster of accounts is false. In the United States of America, Arlington released a study that was devoted to reverse engineering mobile apps. Through the use of a method known as Reverse Engineering Mobile Application User Interfaces (REMAUI), it is performed automatically. Using computer vision and optical character recognition (OCR) methods, REMAUI is able to recognize user interface components such as pictures, words, containers, and lists based on the input that is provided. There were 448 screenshots of Android and iOS programs that were utilized in this system. The user interfaces that were created by REMAUI were quite comparable to the originals, both in terms of their runtime and as far as the pixels themselves were concerned.

C. Krishna B Kansara et al. (2016) This paper proposed a Sybil node discovery method based on the social graph. This approach overcomes the limitations of the previous graph-based approaches by adding user behavioral manners such as latent dealings and friendship refusal. The proposed design is divided into two parts, Sybil node identification (SNI) and Sybil node identification using behavioral analysis (SNI-B)

D. Ali M. Meligy (2017) This paper presents a technique to detect fake accounts on social networking sites called fake profile recognizer. This technique is based on two methods i.e regular expression and deterministic finite automata. A regular expression is used to authenticate the profiles and deterministic automata recognize the identities in a trusted manner.

Samala Durga Prasad Reddy (2019) Used a random forest classifier to detect fake accounts with 95% accuracy. Profile features like id, name, status count, followers count, friends count, location, date of creation of the id, numbers of shares done by the account, gender and language used by the account holder were used as features for the classification process.

Rohit Raturi (2018) Proposed 2 architectures for solving this issue. The first one uses NLP and it marks 2 or more accounts as suspicious if they use the same IP or MAC address. In the second architecture Support Vector Machine(SVM) is used for finding out accounts whichmake frequent use of harmful words. These suspicious accounts then have to verify themselves.

3-SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are :

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on organization. The amount of fund that the company can pour into the research and development of system is limited. The expenditures must be justified. Thus the developed system as well within budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

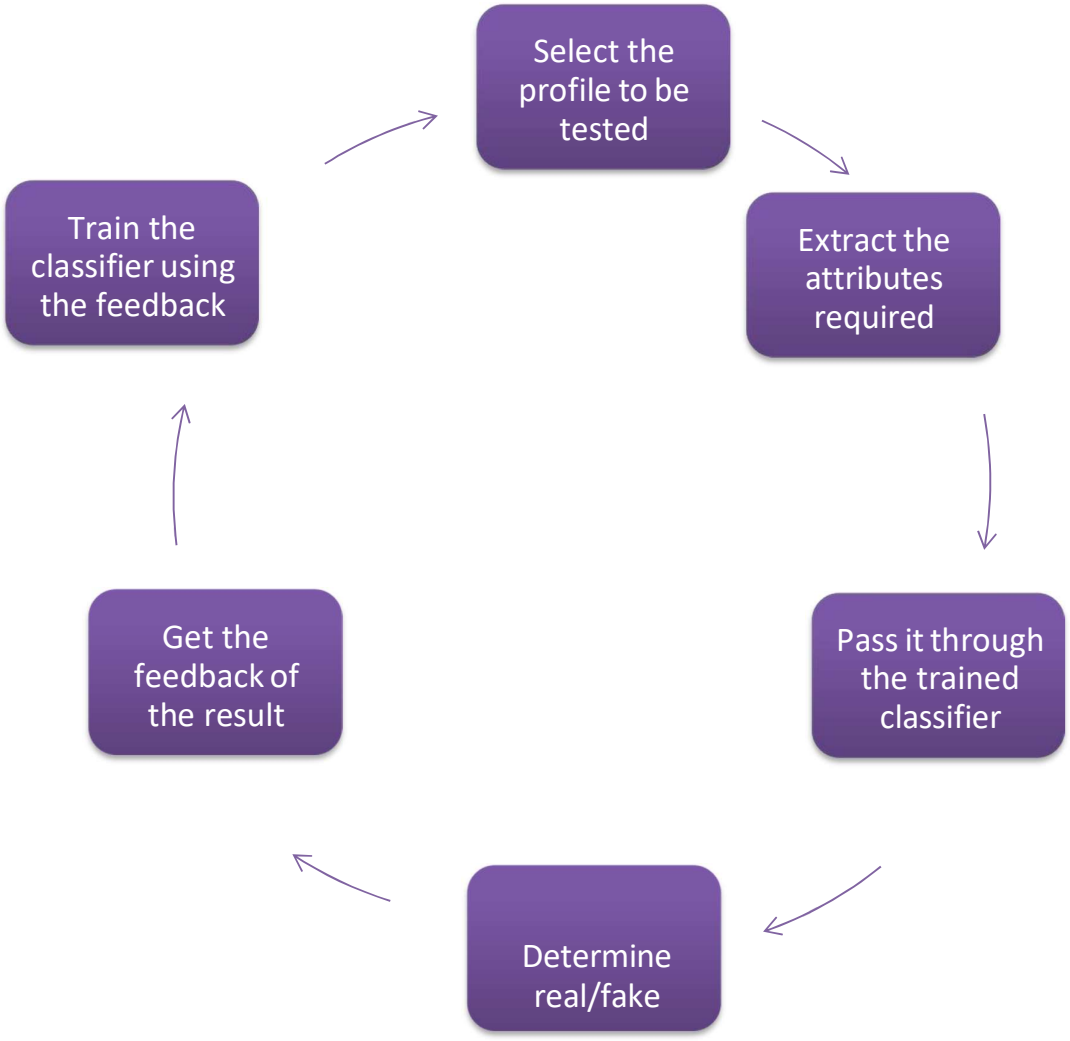
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

UML DIAGRAMS

The acronym UML abbreviates "Unified Modeling Language." In the realm of object-oriented software engineering, the Unified Modeling Language (UML) is a general-purpose modeling language that has been standardized. The Object Management Group is responsible for the management of the standard, as well as its creation. The Unified Modeling Language (UML) is intended to become a standard language for the development of models of object-oriented computer software. A Meta-model and a notation are the two primary components that make up the Unified Modeling Language (UML) in its present form. In the future, the Unified Modeling Language (UML) could also include or be coupled with a method or process of some kind. For the purpose of describing, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other nonsoftware systems, the Unified Modeling Language (UML) is a standard language. A set of best engineering practices that have been shown to be effective in the modeling of big and complex systems is represented by the Unified Modeling Language (UML). Object-oriented software development and the software development process both benefit greatly from the use of the Unified Modeling Language (UML). Graphs are the primary means by which the Unified Modeling Language (UML) expresses the design of software projects.

4-SYSTEM DESIGN

SYSTEM ARCHITECTURE



5-IMPLEMENTATION

PYTHON

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an

implementer on a team building a language called ABC at Centrum Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. Later on in

the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1. Embeddable

   Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

2. Improved Productivity

   The language's need to be in simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

3. IOT Opportunities

   Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

4. Simple and Easy

   When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

6-SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

System Test

System testing ensures that the entire integrated

software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

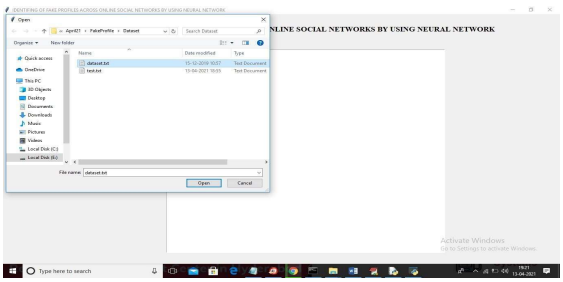
Black Box Testing

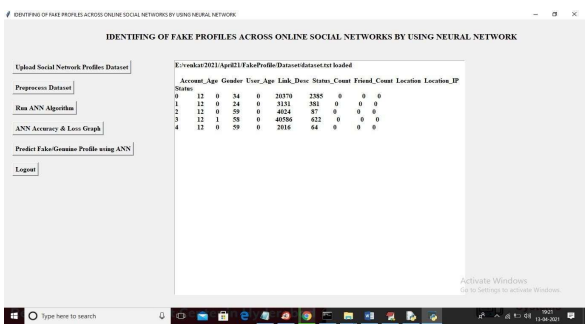
Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

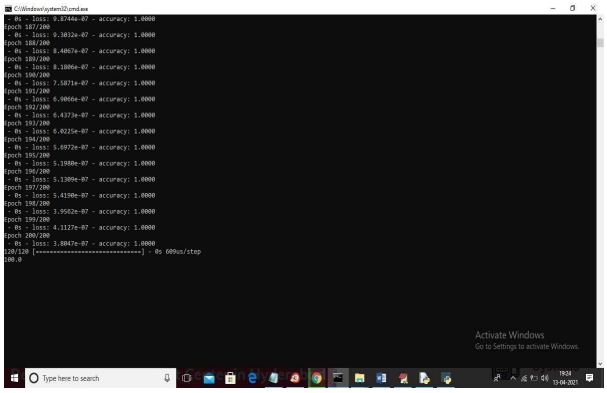
Unit Testing

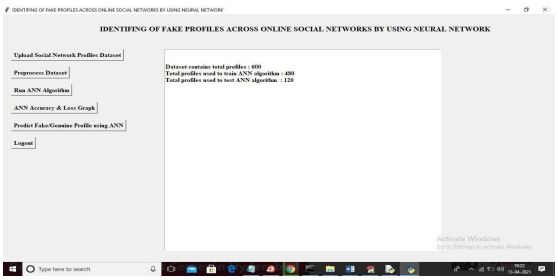
Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7-SCREENSHOTS

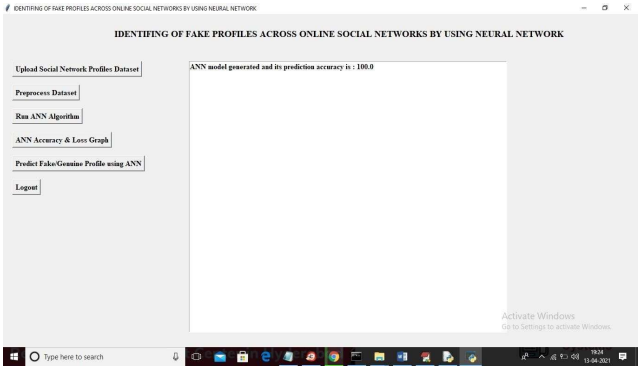





Epoch 187/200
- 0s - loss: 8.8744e-07 - accuracy: 1.0000
Epoch 188/200
- 0s - loss: 9.3032e-07 - accuracy: 1.0000
Epoch 189/200
- 0s - loss: 8.4067e-07 - accuracy: 1.0000
Epoch 190/200
- 0s - loss: 8.1806e-07 - accuracy: 1.0000
Epoch 191/200
- 0s - loss: 7.5871e-07 - accuracy: 1.0000
Epoch 192/200
- 0s - loss: 6.9066e-07 - accuracy: 1.0000
Epoch 193/200
- 0s - loss: 6.4373e-07 - accuracy: 1.0000
Epoch 194/200
- 0s - loss: 6.0225e-07 - accuracy: 1.0000
Epoch 195/200
- 0s - loss: 5.6972e-07 - accuracy: 1.0000
Epoch 196/200
- 0s - loss: 5.1980e-07 - accuracy: 1.0000
Epoch 197/200
- 0s - loss: 5.1309e-07 - accuracy: 1.0000
Epoch 198/200
- 0s - loss: 5.4190e-07 - accuracy: 1.0000
Epoch 199/200
- 0s - loss: 3.9562e-07 - accuracy: 1.0000
Epoch 200/200
- 0s - loss: 4.1127e-07 - accuracy: 1.0000
120/120 [==============================] - 0s 609us/step
100.0
Activate Windows
Go to Settings to activate Windows.
Type here to search


IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout
Dataset contains total profiles : 600
Total profiles used to train ANN algorithm : 480
Total profiles used to test ANN algorithm : 120
Activate Windows
Go to Settings to activate Windows.
Type here to search


Non-trainable params: 0
None
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
Epoch 1/200
- 0s - loss: 10.2115 - accuracy: 0.9396
Epoch 2/200
- 0s - loss: 2.1024 - accuracy: 0.9375
Epoch 3/200
- 0s - loss: 8.8042 - accuracy: 0.9479
Epoch 4/200
- 0s - loss: 5.0597 - accuracy: 0.9604
Epoch 5/200
- 0s - loss: 2.8547 - accuracy: 0.9583
Epoch 6/200
- 0s - loss: 2.6966 - accuracy: 0.9417
Epoch 7/200
- 0s - loss: 9.7434 - accuracy: 0.9542
Epoch 8/200
- 0s - loss: 12.4045 - accuracy: 0.9604
Epoch 9/200
- 0s - loss: 5.1349 - accuracy: 0.9567
Epoch 10/200
- 0s - loss: 1.8743 - accuracy: 0.9521
Epoch 11/200
- 0s - loss: 2.7153 - accuracy: 0.9688
Epoch 12/200
- 0s - loss: 2.8212 - accuracy: 0.9479
Epoch 13/200
- 0s - loss: 1.3293 - accuracy: 0.9583
Epoch 14/200
- 0s - loss: 1.5468 - accuracy: 0.9688
Epoch 15/200
- 0s - loss: 1.8952 - accuracy: 0.9375
Epoch 16/200
- 0s - loss: 1.7365 - accuracy: 0.9604
Epoch 17/200
- 0s - loss: 1.3218 - accuracy: 0.9396
Epoch 18/200
- 0s - loss: 1.2545 - accuracy: 0.9688
Epoch 19/200
Activate Windows
Go to Settings to activate Windows.


IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout
ANN model generated and its prediction accuracy is : 100.0
Activate Windows
Go to Settings to activate Windows.
Type here to search

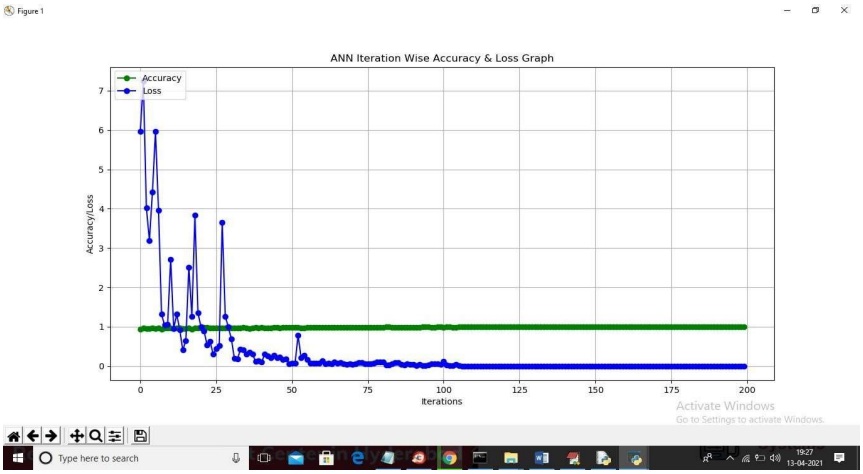
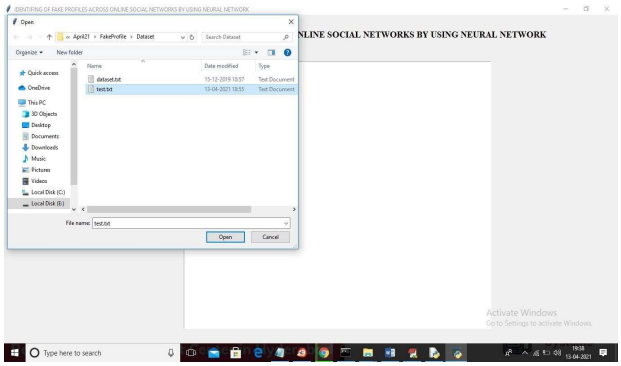
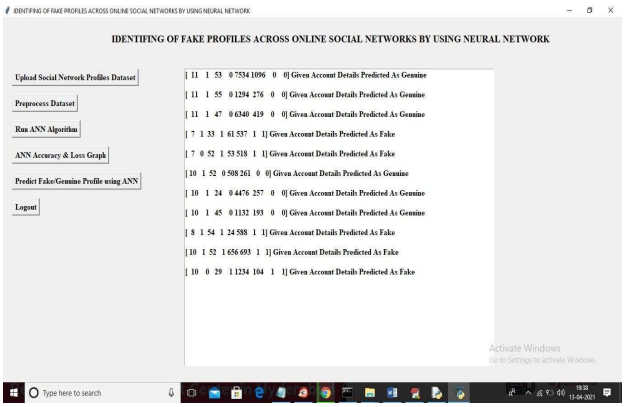


Figure 1
ANN Iteration Wise Accuracy & Loss Graph
Accuracy
Loss
Accuracy/Loss
Iterations
Activate Windows
Go to Settings to activate Windows.
Type here to search


IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
NLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
Open
Search Dataset
Organize
New folder
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
Date modified
Type
dataset.txt
15-12-2019 10:57
Text Document
test.txt
13-04-2021 18:55
Text Document
File name: test.txt
Open
Cancel
Activate Windows
Go to Settings to activate Windows.
Type here to search


IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout
[11 1 53 0 7534 1096 0 0] Given Account Details Predicted As Genuine
[11 1 55 0 1294 276 0 0] Given Account Details Predicted As Genuine
[11 1 47 0 6340 419 0 0] Given Account Details Predicted As Genuine
[7 1 33 1 61 537 1 1] Given Account Details Predicted As Fake
[7 0 52 1 53 518 1 1] Given Account Details Predicted As Fake
[10 1 52 0 508 261 0 0] Given Account Details Predicted As Genuine
[10 1 24 0 4476 257 0 0] Given Account Details Predicted As Genuine
[10 1 45 0 1132 193 0 0] Given Account Details Predicted As Genuine
[8 1 54 1 24 588 1 1] Given Account Details Predicted As Fake
[10 1 52 1 656 693 1 1] Given Account Details Predicted As Fake
[10 0 29 1 1234 104 1 1] Given Account Details Predicted As Fake
Activate Windows
Go to Settings to activate Windows.
Type here to search

CONCLUSIONS

- The Neural Network-Based Detection System holds significant potential to enhance the security of social media platforms by effectively identifying and countering fraudulent profiles
- The goal of building a Neural Network-Based Detection System addresses the urgent need to combat online fraud in social media.
- Its emphasis on ethical compliance and user-friendly design underscores a commitment to security And user satisfaction, marking it as a valuable asset in the ongoing battle against fraudulent activities online.
- This proposed hybrid technique is used to most successful classifier neural network and is also used to improve the accuracy and reduce the time complexity of the algorithm. In proposed work collected realtime data set of Facebook or Twitter from Facebook or Twitter users.

REFERENCES

[1] Awasthi, S., Shanmugam, R., Jena, S.R. and Srivastava, A., 2020. Review of Techniques to Prevent Fake Accounts on Social Media.

[2] Hajdu, G., Minoso, Y., Lopez, R., Acosta, M. and Elleithy, A., 2019, May. Use of Artificial Neural Networks to Identify Fake Profiles. In 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-4). IEEE.

[3] Kaur, J. and Sabharwal, M., 2018. Spam detection in online social networks using feed forward neural network. In RSRI conference on recent trends in science and engineering.

[4] Khaled, S., El-Tazi, N. and Mokhtar, H.M., 2018, December. Detecting fake accounts on social media.In 2018 IEEE International Conference on Big Data (Big Data) (pp. 3672-3681). IEEE.

[5] Meligy, A.M., Ibrahim, H.M. and Torky, M.F., 2017. Identity verification mechanism for detecting fake profiles in online social networks. Int. J. Comput. Netw. Inf. Secur.(IJCNIS), 9(1), pp.31-39.

[6] Ramalingam, D. and Chinnaiah, V., 2018. Fake profile detection techniques in large- scale online social networks: A comprehensive review. Computers & Electrical Engineering, 65, pp.165-177.Wanda, P. and Jie, H.J., 2020. DeepProfile: Finding fake profile in online social network using dynamic CNN. Journal of Information Security and Applications, 52, p.102465.

[7] Zhang, J., Dong, B. and Philip, S.Y., 2020, April. Fakedetector: Effective fake news detection with deep diffusive neural network. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1826-1829). IEEE.

[8] M. Egele, G. Stringhini, and G. Vigna "Towards Detecting Compromised Accounts on social Networks," IEEE, vol. 5971, no. c, 2015.

[9] D. M. Freeman and T. Hwa, "Detecting Clusters of Fake Accounts in Online Social Networks Categories and Subject Descriptors," AISec, 2015.

[10] M. Meligy, "Identity Verification Mechanism for Detecting Fake Profiles in Online Social Networks," IJCNIS, no. January, pp. 31–39, 2017.

[11] Ashraf Khalil, Hassan Hajjdiab, and Nabeel Al-Qirim , Detecting Fake Followers in Twitter: A Machine Learning Approach, International Journal of Machine Learning and Computing, No. 6, December 2017

[12] S. Khaled, N. El-Tazi and H. M. O. Mokhtar, "Detecting Fake Accounts on Social Media," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp.