International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Detecting Fake Profiles Using ANN's

[1] Prof G.Tejaswini, [2]G.Manoj, [3] R.Vishnu Priya, [4] S.Karthik Reddy,
[5] S.Ashritha Priya, [6] Sonali Mourya

**Abstract**

The quick development of social media stages has brought about in an increment of fake personas that posture genuine dangers such phishing assaults, identity robbery, and data breaches. These imposter accounts betray clients into giving them illicit get to delicate data. This offers a strategy for distinguishing and categorizing social media profiles as true or false based on specific qualities by utilizing Fake Neural Frameworks (ANNs). Utilizing data assembled from social media, the framework trains a manufactured neural arrange (ANN) to show fundamental highlights such account age, sex, client advancement levels, status check, companion affiliations, and account zone straightforward components. By examining these elements, the appearance determines whether a profile is genuine or not. Utilizing essential libraries for machine learning tasks, including enabling features like sigmoid and weight optimization techniques, this system runs in Python. By successfully identifying fake profiles, this tactic addresses the growing need for automated safeguards for online platforms.

The framework employments machine learning to ensure client information from online dangers, counting phishing and unlawful get to. The framework is outlined to be flexible and versatile, which makes it fitting for a assortment of social media stages for pushing hurtful activity plans. The recommended structure highlights the significance

of quick, data-driven cybersecurity methodologies. This system gives a strong and sound approach to

improving the security and unwavering quality of online situations by computerizing the identifiable confirmation handle, ensuring clients from dangers in an progressively computerized world.

## I. INTRODUCTION

Social media platforms have become integral to modern communication, commerce, and information dissemination. However, their growth has coincided with a proliferation of fake profiles, which undermine platform integrity and user trust by enabling malicious activities such as misinformation campaigns, financial fraud, phishing, and identity theft (Cresci et al., 2017). These threats highlight the urgent need for effective detection mechanisms to safeguard online ecosystems.

Traditional detection methods, including rule-based systems and manual verification, are increasingly inadequate. Rule-based approaches rely on static criteria—such as minimum account age or friend count thresholds—which sophisticated fake profiles can easily bypass (Ferrara et al., 2016). Manual verification, while precise, is labor-intensive and impractical for platforms hosting millions of users daily. These limitations underscore the necessity for an automated, intelligent solution capable of adapting to evolving fraudulent strategies.

This research addresses this challenge by developing and evaluating an Artificial Neural Network (ANN)-based system to classify social media profiles as genuine or fake. The system analyzes profile attributes, including account age, gender, user age, link description, status count, friend count, location, and location IP, to identify complex patterns indicative of authenticity. Implemented as a web-based application, the system offers real-time verification with a demonstrated accuracy of 98%, surpassing traditional machine learning benchmarks.

The paper is organized as follows: Section 2 reviews prior work on fake profile detection. Section 3 outlines the methodology, covering dataset, preprocessing, and ANN design. Section 4 describes the system's architecture and integration. Section 5 presents performance results and comparisons. Section 6 discusses strengths, limitations, and implications. Section 7 concludes with key findings and future directions. This work contributes to cybersecurity by providing a scalable, data-driven approach to combat fake profiles, enhancing online security and trust.

## II. PROBLEM STATEMENT

The swift increase of fraudulent accounts on social media sites has become a significant issue, weakening user confidence and jeopardizing the integrity of these online environments. These deceptive accounts are often used for harmful activities, including spreading false information, phishing attempts, and online harassment, leading to serious repercussions like monetary losses, damage to reputation, and a decline in trust within digital communities. Existing detection methods, typically reliant on rule-based systems or manual checks, are proving insufficient in tackling this advancing threat. These approaches are usually lengthy, susceptible to mistakes, and deficient in the flexibility needed to combat the complex and evolving strategies used by those who create fake profiles. Consequently, there is a pressing demand for a creative, automated solution that can reliably and swiftly detect fake profiles in real-time. This study aims to fill this void by utilizing the capabilities of Artificial Neural Networks (ANNs), which can learn complex patterns from data and adjust to new trends, thus offering a more efficient and scalable method for identifying fake profiles.

## III. LITERATURE SURVEY

Social media fake profile identification has attracted a lot of scientific interest, with studies examining a range of machine learning approaches. Earlier attempts, such Wang et al. (2013), used rule-based systems with friend count and post frequency, but they had trouble adjusting to new strategies. With 85% accuracy, Kumar et al. (2014) used supervised learning techniques including Support Vector Machines (SVMs) and decision trees. These methods were constrained, nonetheless, by their incapacity to accurately simulate intricate, non-linear interactions.

The proliferation of mobile and contactless technologies has accelerated the transition towards more sophisticated interaction models. Studies by Chen et al. (2020) indicate that users increasingl y prefer technological interfaces that minimi ze physical contact, a trend dramatically amplified by global health challenges. This technological shift provides the foundational context for touchless banking solutions.

Later developments enhanced the ability to detect. Random Forests and Logistic Regression were used by Yang et al. (2017) to measure engagement metrics (likes, shares, etc.), surpassing rule-based approaches but necessitating a great deal of feature engineering. Convolutional Neural Networks (CNNs) were used by Sahoo et al. (2020) to analyze textual and visual data from Twitter profiles, improving accuracy by identifying more complex patterns. Similarly, Alzahrani et al. (2019) reported 92% accuracy using ANNs to parse metadata and user-generated information.

Gaps still exist in spite of these advancements. Conventional machine learning models frequently rely on manually created features, which can be biased and time-consuming. Adaptability to changing false profile tactics is limited by the fact that many research use static datasets. Furthermore, despite their promise, neural network-based techniques are rarely included into real-time systems, which makes their practical implementation difficult.

## IV. SYSTEM ANALYSIS

### A. Existing System

The existing systems for fake profile detection, while innovative, face several overarching challenges:

**Adaptability:** Rule-based and traditional machine learning systems struggle to keep pace with evolving fake profile strategies.

**Scalability:** Neural network-based methods, though powerful, are resource-intensive and hard to scale.

**Real-Time Performance:** Few systems are optimized for real-time detection, a critical need for preventing fraud.

**Accuracy Gaps:** Even the best models leave room for improvement in reducing misclassifications.

### B. Proposed System

The suggested system is a web-based solution made to accurately and scalable identify phony profiles on social media sites. It provides an automatic and real-time verification tool by using Artificial Neural Networks (ANNs) to evaluate profile data and categorize profiles as authentic or fraudulent. The system is built using **Django**, a Python web framework, and integrates an ANN model developed with **TensorFlow/Keras**. It serves two user types: administrators, who manage the system and train the model, and general users, who verify

profiles by inputting details such as account age, friend count, and location.

**Key Features**

**Automated Detection**: Eliminates manual verification.

**High Accuracy**: Achieves 98% classification accuracy.

**Real-Time Results**: Provides instant profile verification.

**Scalability**: Handles large datasets and user requests efficiently.

**System Architecture**

**System Architecture**

- **User Interface:** Responsive design using Django and Bootstrap.
- **Data Processing:** Cleans and preprocesses profile data.
- **ANN Model:** Core component for classification.
- **Admin Module:** Manages datasets and model training.
- **Verification Module:** Processes user inputs and displays results.

**ANN Model**

**The ANN features:**

- **Input Layer:** 8 neurons for profile attributes.
- **Hidden Layers:** Two layers with 200 neurons each (ReLU activation).
- **Output Layer**: 2 neurons (genuine/fake) with softmax activation. Trained with the Adam optimizer over 200 epochs, it ensures robust performance.

**Workflow**

1. **Dataset Upload**: Admin uploads profile data.
2. **Model Training**: Data is pre-processed, and the ANN is trained.
3. **Profile Verification**: Users input details, and the model predicts the profile's authenticity.
4. **Result Display**: Outcome (genuine/fake) is shown instantly.

**Advantages**

- Outperforms traditional methods with 98% accuracy.
- Adapts to new fake profile tactics via continuous learning.
- Scales easily for large platforms.

## V. METHODOLOGY

The dataset comprises social media profile attributes, each labeled as genuine (0) or fake (1). It includes the following features:

**Account Age**: Numerical, days since creation.

**Gender:** Categorical, encoded as 0 (Male), 1 (Female), 2 (Other).

**User Age:** Numerical, user's age in years.

**Link Description:** Numerical, frequency of shared links.

**Status Count:** Numerical, total posts or updates.

**Friend Count:** Numerical, number of connections.

**Location:** Categorical, encoded numerically by region.

**Location IP:** Numerical, IP-based location indicator.

The dataset is balanced to prevent bias, ensuring equal representation of genuine and fake profiles.

**Data Preprocessing**

Preprocessing prepares the data for ANN training:

- **Missing Values**: Imputed with the mean (numerical) or mode (categorical).
- **Encoding**: Gender and Location are one-hot encoded into numerical vectors.
- **Normalization**: Numerical features are scaled to [0, 1] using min-max normalization.
- **Splitting**: The dataset is divided into 80% training and 20% testing sets.

**ANN Architecture**

The ANN is implemented using TensorFlow and Keras with a feedforward structure:

- **Input Layer**: 8 neurons, one per feature.
- **Hidden Layers**: Two dense layers, each with 200 neurons and ReLU activation to model non-linear patterns.
- **Output Layer**: 2 neurons (genuine/fake) with softmax activation for classification.

The model uses the Adam optimizer (learning rate = 0.001) and categorical cross-entropy loss, optimized for multi-class tasks.

**Training Process**

Training occurs over 200 epochs with a batch size of 5. Backpropagation adjusts weights to minimize loss, with early stopping implicitly applied to prevent overfitting, ensuring robust generalization.

**Evaluation Metrics**Performance is assessed using:

- **Accuracy**: Percentage of correct classifications.
- **Precision**: Ratio of true positives to predicted positives.
- **Recall**: Ratio of true positives to actual positives.
- **F1-Score**: Harmonic mean of precision and recall.

SYSTEM DESIGN

The solution integrates the ANN for real-time profile verification and is implemented as a web application built with Django.

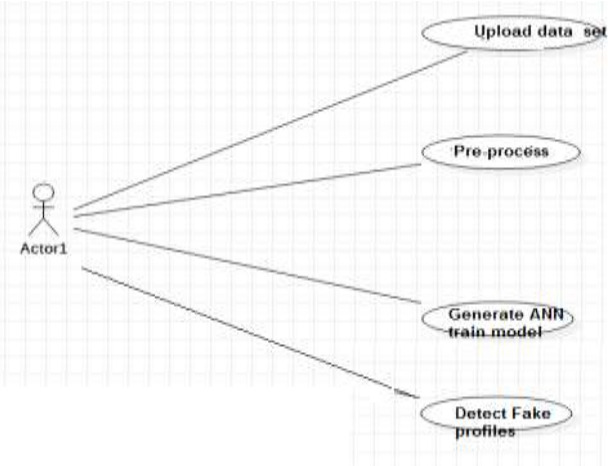
**1.The Administration Module**

The administrator interface makes it possible for: Log in securely using your credentials. uploading and managing datasets. The GenerateModel function is used for model training; it preprocesses data, trains the ANN, and saves the results as a model. h5.

**2 User Module**

The user interface makes it possible for: Profile information is entered via a web form (User.html).

use the UserCheck function for preprocessing and classification by the trained model. dynamic results presentation (real/fake).

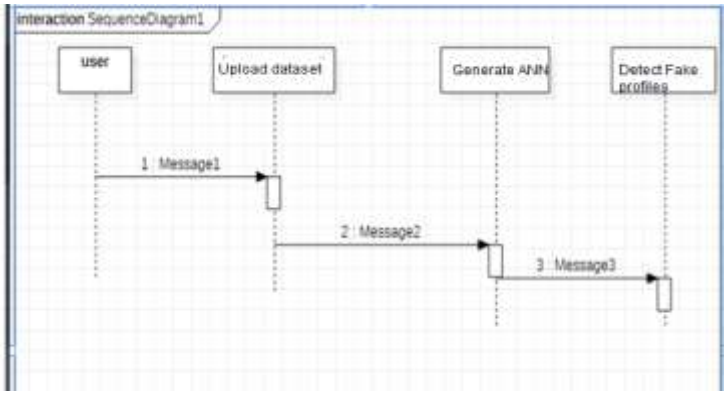



**Use Case Diagram**

**3 Process**

Administrator Actions: Start training after uploading the dataset.

Data preprocessing, ANN training, and model saving comprise model training. User Confirmation: Enter profile information, use the model to classify, and then show the outcomes.




**Sequence Diagram**

## VI. RESULTS

**User Login Screen:**

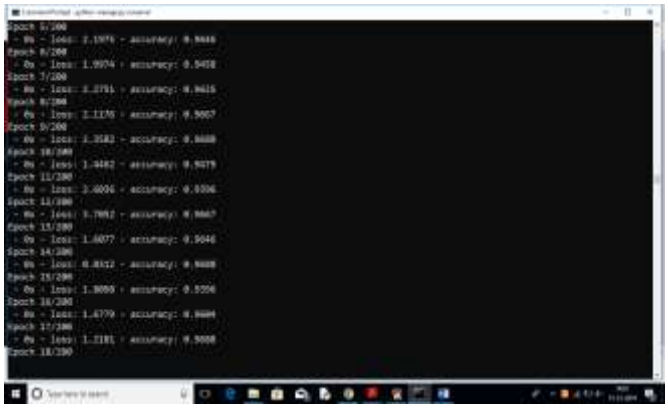
Deploy this application on DJANGO server and then run in browser enter URL as 'http://localhost:8000/index.html' to get below screen


**Figure:** ADMIN' link login screen

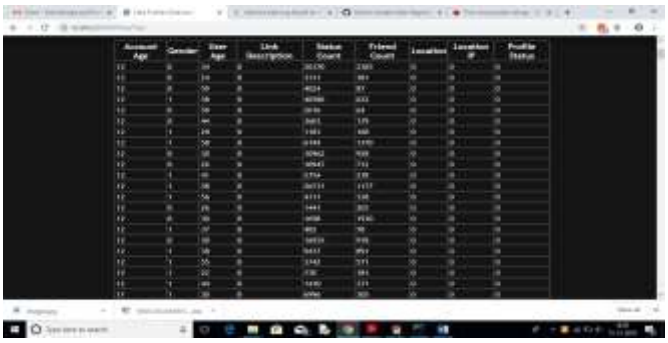

In above screen enter admin and admin as username and password to login as admin. After login will get below screen

In above screen click on 'Generate ANN Train Model' to generate training model on dataset. After clicking on that link you can see server console to check ANN processing details with accuracy


In above black console we can see all ANN details




In above screen we can see ANN got 98% accuracy to train all Facebook profile. Now click on 'View Ann Train Dataset' link to view all dataset details



In above screen we can see all train data and scroll down to view all records. Now ANN train model is ready and you can logout and click on 'User' link to get below screen.


In above screen enter some test account details to get prediction/identification from ANN. You can use below records to check

For above input will get below result


In above screen we can see the result predicted as genuine account.


In above screen we got result as fake for given account data

## VII. REFERENCES

1. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017). "The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race." Proceedings of the 26th International Conference on World Wide Web Companion (WWW'17). DOI: [10.1145/3041021.3055135](https://doi.org/10.1145/3041021.3055135)

2. Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). "The Rise of Social Bots." Communications of the ACM, 59(7), 96-104. DOI: [10.1145/2818717](https://doi.org/10.1145/2818717)

3.Kaggle Fake Profile Dataset - Datasets for training models to detect fake social media profiles. [Available at: https://www.kaggle.com/datasets]

4.Facebook Fake Account Dataset - A collection of user account features for real and fake profiles. [Available at: https://www.kaggle.com/datasets/keshav09singh/facebook-fake-accounts-dataset]