Fake User Detection using Artificial Neural Networks and NLP-based Anomaly Detection

Author Name(s)  
Department Name, Institution Name, City, Country  
Email Address

# Abstract

The proliferation of fake accounts on social media platforms is a growing issue that impacts user experience and platform integrity. This paper presents a solution for fake user detection using a combination of Artificial Neural Networks (ANN) and Natural Language Processing (NLP)-based anomaly detection techniques. The system is designed to classify users as either real or fake based on several features, including user activity data and bio content. Additionally, anomaly detection methods such as Isolation Forest are employed to detect outliers in the text features of the user bio, providing additional information to enhance the classification accuracy. The model is trained on real and fake user data and achieves high accuracy in identifying fraudulent accounts.

# 1. Introduction

Fake user accounts are a persistent problem in online communities, including social media platforms, e-commerce websites, and forums. These accounts often engage in malicious activities such as spam, identity theft, and spreading misinformation. Traditional methods of detecting fake accounts rely on basic heuristics or rule-based systems, which are often inadequate in dealing with sophisticated fake account behaviors.  
In this paper, we propose a machine learning approach for fake user detection by using a combination of Artificial Neural Networks (ANN) for classification and Natural Language Processing (NLP)-based anomaly detection for enhancing model performance. We utilize user metadata (e.g., statuses count, followers count) and textual features (user bios) to detect suspicious or fraudulent user activity.

# 2. Methodology

The proposed system is designed to perform fake user detection using both structured and unstructured data. The system includes the following key components:  
1. Data Collection and Preprocessing: We obtain data from two sources: real user accounts and fake user accounts. The data includes user metadata (such as the number of statuses, followers, and friends) as well as textual content (the bio or posts). The bios are preprocessed using NLP techniques like TF-IDF vectorization to convert text into numerical features.  
2. Anomaly Detection with NLP: Anomaly detection algorithms such as Isolation Forest are applied to the processed text features (bios) to identify unusual patterns. This helps in capturing outliers, which might indicate a fake user.  
3. Artificial Neural Network (ANN) for Classification: An ANN model is used to classify the users based on both the numerical and text features. The network consists of multiple hidden layers, with the final output being a binary classification (real or fake).  
4. Model Training: The model is trained on a labeled dataset, where the users are categorized as real or fake. The performance of the model is evaluated using standard metrics like accuracy, precision, recall, and F1 score.  
5. Prediction and Anomaly Score: For new users, the trained model predicts whether they are real or fake based on their metadata and bio content. Additionally, an anomaly score is generated based on the text features to provide extra information about the user's authenticity.

# 3. Technologies Used

The system utilizes the following technologies:  
• Python: The primary programming language for implementing the solution.  
• Keras/TensorFlow: For building and training the Artificial Neural Network model.  
• Scikit-learn: Used for preprocessing the data, applying anomaly detection techniques, and evaluating the model.  
• Gradio: For creating an interactive web-based interface that allows users to upload files and view predictions.  
• Matplotlib: Used for visualizing the training metrics (accuracy and loss).  
• Isolation Forest: A machine learning algorithm used for anomaly detection in the bio text data.  
• TfidfVectorizer: A method from Scikit-learn used for converting text (bio) into numerical vectors.

# 4. System Requirements

• Hardware Requirements:  
 - Processor: Intel Core i5 or equivalent (minimum)  
 - RAM: 8 GB or more  
 - Storage: 10 GB of free space for storing models, datasets, and logs  
• Software Requirements:  
 - Python 3.x  
 - TensorFlow 2.x  
 - Keras  
 - Scikit-learn  
 - Pandas  
 - Matplotlib  
 - Gradio  
 - Jupyter Notebook (for running code and visualizations)  
• Operating System:  
 - Windows 10 or higher  
 - Linux (Ubuntu 20.04 or higher)  
 - macOS 10.14 or higher

# 5. Diagrams

• System Architecture Diagram:  
 +-------------------------+  
 | User Interface |  
 | (Gradio Web App) |  
 +-------------------------+  
 |  
 v  
 +-------------------------+  
 | Data Preprocessing |  
 | (Text Preprocessing, |  
 | Feature Engineering) |  
 +-------------------------+  
 |  
 v  
 +-------------------------+  
 | Anomaly Detection |  
 | (Isolation Forest) |  
 +-------------------------+  
 |  
 v  
 +-------------------------+  
 | Neural Network Model |  
 | (ANN for Classification)|  
 +-------------------------+  
 |  
 v  
 +-------------------------+  
 | Prediction Output |  
 | (Fake/Real & Anomaly) |  
 +-------------------------+

# 6. Conclusion

In this paper, we have demonstrated an effective approach to detecting fake user accounts using a combination of Artificial Neural Networks (ANN) and NLP-based anomaly detection. By incorporating anomaly detection techniques such as Isolation Forest into the feature engineering process, we can enhance the model's ability to identify unusual patterns that might indicate a fake user. The model performs well on real and fake user data and provides an interactive interface for making predictions.  
The proposed system is flexible and can be extended to incorporate more advanced anomaly detection algorithms or different classification models. This research provides an important step forward in addressing the problem of fake user accounts, with potential applications in social media, e-commerce, and other online platforms.

# 7. Future Enhancements

While the current system shows promising results, there are several ways it can be enhanced:  
1. Advanced NLP Techniques: Integrating pre-trained language models such as BERT, GPT, or Word2Vec for better text feature extraction.  
2. Real-Time Detection: Implementing a real-time system for monitoring and detecting fake users in online platforms.  
3. Hybrid Models: Combining ANN with other models, such as decision trees or random forests, for more robust predictions.  
4. Explainable AI (XAI): Incorporating explainability methods to provide insights into why a user is predicted to be fake.  
5. Integration with Other Data Sources: Leveraging additional data, such as user activity or network data, to improve detection accuracy.