



Supervised Machine Learning Based Fake Profile Detection Using User Ratings and Reviews in Recommender Systems

Ümmügülsüm Mengutaycı¹ , Selma Ayşe Özel² 

Article Info

Received: 18 Mar 2025

Accepted: 28 Jun 2025

Published: 30 Jun 2025

Research Article

Abstract – Recommendation systems produce content based on user's interests and aim to increase user satisfaction. In this way, the system keeps the user constantly active. Therefore, the reliability and robustness of these systems are essential. However, in recent years, with the influence of popular culture, recommendation systems have been struggling with fake users to highlight a particular product more or, conversely, to reduce the popularity of the product. Fake accounts mimic real user data and provide misleading information to the systems. This affects the accuracy of recommendation algorithms. This paper proposes a novel approach to detect fake user profiles by combining two different data sources: rating data and product reviews by using machine learning techniques, such as Decision Trees, Logistic Regression, Support Vector Machines, k-Nearest Neighbors and Naive Bayes algorithms. We also test the impact of integrating ensemble learning techniques on classification success. The research results show that the ensemble learning method Stack Classifier model has the highest detection success with an F1-score of 81.11%. This highlights that the innovative approach using multiple data sources together provides a more robust and reliable solution for detecting fake profiles, thus improving the accuracy and efficiency of recommender systems.

Keywords – Recommender system, fake profile detection, machine learning, robustness

1. Introduction

With the development of technology and the internet, people have started to spend more time on social media today. In these social platforms, which can even be downloaded to mobile phones, users can easily express their feelings and thoughts, and users can influence each other. A user who wants to receive a service examines and analyzes the opinions of other users who have previously used the same service or product. For this reason, digital platform owners will aim to bring them together with the right service to avoid losing their customers. These platforms develop special content for their customers using recommendation algorithms. Recommendation systems are essential in increasing user satisfaction and participation by offering content adapted to personal preferences. Therefore, the accuracy of the data entered into these systems is crucial for their effectiveness and reliability. However, in the competitive market, especially in recent years, the number of fake user accounts on online platforms has increased. These fake accounts in the system use real user's data by imitating them and injecting misleading information into the system. This situation negatively affects the outputs of recommendation algorithms. As a result, user trust decreases, and customer dissatisfaction increases [1].

¹mengutayci@tarsus.edu.tr (Corresponding Author); ²saozel@cu.edu.tr

¹Department of Computer Engineering, Faculty of Engineering, Tarsus University, Mersin, Türkiye

²Department of Computer Engineering, Faculty of Engineering, Çukurova University, Adana, Türkiye

Fake users evaluate and interpret the service/product in a way that suits their purposes to direct real user behavior to their desired goal. In this way, they mislead recommendation algorithms. Over time, a real user in the system may be matched with an unsuitable service, which may cause unpleasant situations. Consumers who are not satisfied with the service they receive will come to the point of completely stopping using the platform in the long run [1, 2]. The detection of fake users can often be assumed to be a binary classification problem. Binary classification divides users into fake and real users [1]. Supervised machine learning techniques effectively perform this detection, which is a binary classification problem.

Numerous traditional approaches [3 - 9] depend on only a single data source to detect fake users, for example, focusing only on analyzing products that users have rated. Given the ever-increasing mass of data [1], a more comprehensive analysis is required to identify fake profiles. In some recent studies [10-14] it has been observed that using ratings, user reviews, and user characteristics together improves the classification success in detecting fake users. Based on this inspiration, combining two different data sources, such as rating data and item reviews, can give a more effective solution to ensure the accuracy of recommendation systems and reduce the impact of fake users. Rating data reflects the user's rating and appreciation of the product, while product reviews reveal the user's thoughts and experiences. Based on these two different data sources, detecting fake users will be much more robust and reliable.

In this study, rating data and product reviews will be used together to distinguish fake users, and these attributes will be classified using machine learning algorithms. Machine learning is a powerful tool for detecting patterns in large and complex datasets. To detect fake users, inconsistencies between the ratings given by users and the comments they write or deviations from the norm can be highlighted by classification algorithms. In this context, support vector machines (SVM), decision trees (DT), logistic regression (LR), k-nearest neighbors (kNN) and naive bayes (NB) machine learning algorithms, which have shown high success in detecting fake users, will distinguish fake users from real users [15-18]. Unlike previous studies, this innovative approach combines data sources to ensure accurate and efficient detection of fake profiles. In addition, the research utilizes ensemble learning techniques and combines the outputs of machine learning algorithms. The outputs are given to a second model, the XGBoost algorithm and the classification success is tested. Thus, the effectiveness of the two methods in detecting fake users in the system is compared.

2. Related Works

Previous research on fake user attacks in recommender systems involves statistical and machine learning methods. The authors in [4] combined statistical and machine learning methods. Statistical techniques detect anomalies in user behavior, and machine learning is applied to classify them as real or fake users. In most recommender systems, the number of labeled users is limited, and the number of unlabeled users is usually large, and labeling large amounts of data will be costly. In [9], the authors proposed the Semi-supervised learning based Shilling Attack Detection (Semi-SAD) algorithm, a semi-supervised learning approach, to detect fake users. This algorithm initially trains a Naive Bayes classifier on labeled users. Then, it optimizes the classifier by including unlabeled users in the system using the weight factor λ added Expectation Maximization (EM- λ) algorithm. Experiments on the MovieLens dataset show that the proposed approach is more efficient than other methods.

In another research, the authors [19] emphasized the class imbalance problem in supervised detection methods and stated that the detection performance decreases when the number of attack profiles is small. Therefore, they proposed a new method, the Support Vector Machine-Target Item Analysis (SVM-TIA) model, to detect attacks. In the first stage, they solved the class imbalance problem using the Borderline - Synthetic Minority Over-sampling Technique (SMOTE) method. In the second stage, target product analysis is performed with a fine-tuning that examines attack profiles and reduces false positives. In this stage, the authors first used the user rating data in the rating matrix as feature data and performed classification with the SVM algorithm. In this way, the first detections were identified. Target item analysis was then performed on the data labeled as a

potential attack profile based on the classification result. In this way, incorrectly categorized users were weeded out. Since attack profiles will rate the target item according to the attack intent (min or max), the number of evaluations on the target item will be higher than other products.

Researchers [3] have used an ensemble learning approach to eliminate fake users in the system. They have used various classification algorithms, such as Quadratic Discriminant Analysis (QDA), Naive Bayes Classifier (NBC), and kNN, to identify fake profiles and combined these algorithms with the ensemble voting (VE) approach. They emphasized that the ensemble voting technique achieved high accuracy at the end of the study. The authors [20] proposed the Single-Class SVM (OCSVM) method, which is generally used to detect outliers in fake user detection. This new approach builds models using only real user data. In this way, the authors aimed to minimize the need for labeled data to detect attacks. The hyperparameters of the OCSVM algorithm were determined by the Quick Model Selection technique to ensure the model's optimal performance and speed up the selection process. The researchers concluded that OCSVM provides a more applicable method for detecting fake profiles and stated that their research differs from other methodologies in the existing literature on this subject. In [21], authors tried to find a general solution for recommender systems that can detect any attack regardless of its characteristics. For this purpose, they used feedback from verified real users and trained classifiers. The proposed new method is based only on the behavioral characteristics of legitimate users. The authors considered any abnormal behavior in the system as an attack. It was concluded that this innovative approach based on positive unlabeled learning (PU) and single-class SVM (OSVM) models successfully detected unknown attacks.

In the studies mentioned above, a single data source (rating) data was used to detect fake users. However, a user not only rates a product but also provides a review of the product. The authors [15] used machine learning techniques to detect fake reviews on the YELP dataset by incorporating the length of reviews, the maximum number of reviews per user per day, and the average review rating deviation rate into the review data. Another study [22] combines product reviews with some behavioral characteristics of users (total number of capital letters, punctuation marks and emoji) to detect fake users using machine learning techniques. The experimental results show that the XGBoost algorithm provides the highest detection rate. Chopra & Dixit [10] used ratings and reviews together to detect fake users in recommender systems. In the study, the authors labeled product reviews with a score of 1 and 5 as fake. The results of the classification using a recurrent neural network (RNN) and the Bald Eagle Search (BES) optimization algorithm showed that the combination of ratings and reviews increased the detection of fake users. They [11] proposed a new approach to detect fake users using product reviews, aspect ratings (especially ratings based on things like cleanliness, location, service) and overall ratings. After extracting features from product reviews with Bidirectional Encoder Representations from Transformers (BERT), the authors aimed to extract feature vectors with deeper meaning from text data using Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture. They used a fully connected layer to classify the product reviews, where they selected the most important features relevant to the aspect, then combined them with aspect rating and overall rating. The obtained F1 score of 96% shows that the combination of ratings and user reviews is successful in detecting fake users.

The authors [12] proposed a graph-based method to detect spam users. In this method, reviews form the nodes and edges are formed by reviews made by the same user on the same item, reviews made by users who gave the same rating for the same item, and reviews made on the same item in the same month. After the correlation between the comments was determined by using Graph Neural Network (GNN) and Multi-Layer Perceptron (MLP) together, the outputs were given to CNN and LSTM architectures respectively and fraud detection was achieved with high accuracy. Noting that rating data alone does not directly reflect user characteristics, the authors [13] used the information obtained from the user rating matrix, as well as user's rating biases, rating time intervals, review text lengths, and similarity of review texts, to detect spam users. To learn the relationships between users, they used a Graph Sample and Aggregate (GraphSAGE) based method to distinguish between real and fake users. They used an attention mechanism to calculate the contribution of

each node's neighbors. The experimental results show that the combination of the two data types improves the performance of fake user detection. Therefore, we use both rating and review data in our research. Since fake user detection is a binary classification problem, we will identify potential attack profiles using machine learning and ensemble learning methods used in the literature and perform adequately in this field.

3. Method

3.1. Dataset

We used the Yelp dataset [23] from Kaggle for this study. The dataset contains user-review data about restaurants on the Yelp platform related to products and services. The dataset comprises reviewID, reviewerID, rating, reviewContent, and flagged information that indicates whether the user is real or fake. The dataset contains 751232 real user data and 8301 fake user data. The ratings provided by users range from 1 to 5. In the dataset, one is the lowest, and five is the highest. The reviews are written in English.

The dataset consists of the following basic features:

- i. reviewerID: A unique identifier assigned to each user in the dataset.
- ii. reviewID: A unique identifier assigned for each product or service reviewed.
- iii. rating: The rating given to restaurants by users, ranging from 1 to 5.
- iv. reviewContent: The user's opinion about the service they received.
- v. flagged: A label indicating whether the user is genuine or an attacker.

3.2. Dataset Preparation

In the dataset, fake users are a minority class. Due to class imbalance in the dataset, 9,300 real user data were used. Details of the users in the dataset are presented in Table 1. Around 36 percent of the users in the dataset have reviewed and rated only one restaurant. For this reason, the rating matrix is sparse. A rating matrix was created using user, item, and rating data in the classification performed on the rating data. In this matrix, since not all users can evaluate all products, the Nan rating values were replaced with 0.

Table 1. Dataset overview

Data count of real users	9300
Data count of fake users	8301
Number of unique real users	4233
Number of unique fake users	7114

Since we used product review and rating data for fake user profile detection in this study, preprocessing was performed on the textual part of the review data. During preprocessing steps, punctuation marks, special characters, and numeric values were removed from the text. Words that do not affect the meaning, such as prepositions and conjunctions, were filtered out, sentences were tokenized, and lemmatization was applied to the words. There may be more than one product purchased by existing users in the system. Hence, users may review varying numbers of products differently from each other. As textual data is highly dimensional, and each review may have varying words, it is difficult to represent reviews as a fixed-sized matrix as in the rating matrix. Word2Vec was used to overcome this difficulty for word-based vector representation in our research. We employed the Word2Vec model for word embedding and tokenization to handle the textual data in the reviews. Word2Vec [24] is a shallow, two-layer neural network model used to learn vector representations of words in a continuous vector space based on their surrounding context in the text. The idea behind Word2Vec is that words that appear in similar contexts have similar meanings, which are captured through dense vector representations, unlike traditional one-hot encoding that produces sparse vectors.

Word2Vec works on two main architectures:

- i. Continuous Bag of Words (CBOW): This model looks at the words around the target word to predict it.
- ii. Skip-gram: This model works in reverse to the CBOW model and predicts the surrounding words by looking at the target word.

Both models position similar words close together in vector space. These dense word embeddings capture semantic relationships; words with similar meanings or usages will have similar vector representations.

In this study, the vector of each word in the review is found using the Word2Vec model. Then, these vectors are averaged to find the vector of the review. All word vectors in a review were collected for the review vector representation, and their average value was taken to create a 100-dimensional review vector. In such a case, a user's comment about a product would be multidimensional (100-dimensional vector). In user-item review matrix notation where rows represent users and columns represent items, computing such a matrix would be costly and time-consuming, as each matrix element is a 100-dimensional vector. To overcome this problem, we first categorized the product reviews of each uniquely identified user separately and then stored the user information. When making predictions using the test data, we also considered which user the product review belonged to. For example, let's assume that user X in our test dataset has reviewed three different items, and the classification algorithm predicted that the first and second reviews of the user were fake while the third review was genuine. The probability of the classification algorithm for each class prediction was calculated and averaged over all predicted class labels.

The same training and test data were used in rating- and review-based classification. The dataset was split into 80% training and 20% test data. The division into training and test datasets was performed entirely at the user level. After checking that the training and test dataset did not contain the same user ID, the classification step was started.

3.3. Classification

In this section, we describe the algorithms used in classification. A decision tree is a framework that graphically displays the likelihoods and results of a sequence of events or choices. Within this framework, each node signifies an event or decision juncture, while the edges illustrate how these decisions are reached or the factors that affect these choices. In simpler terms, each node reflects an assessment of a scenario or characteristic, while the edges denote the alternatives or routes to pursue based on that scenario.

Naive Bayes is widely used in problems such as text classification. This algorithm calculates the probability between classes and features on training data. It then uses these probabilities on test data to predict which class the new data belongs to.

SVM is a method that establishes a boundary to identify the class to which the data is assigned. This boundary maximizes the margin between the data points, reducing classification mistakes. When the data can be separated linearly, SVM creates a simple linear boundary; however, if the data is intricate and cannot be separated linearly, the kernel trick is employed to convert the data into a higher-dimensional space, where linear boundaries can be applied for classification. This capability makes SVM a highly effective and adaptable classifier.

The kNN algorithm is one of the supervised machine learning algorithms. This algorithm is used in both classification and regression problems. It determines the class or value of a data point by looking at its k nearest neighbors where k is a specified integer value [25].

LR is a probability-based method used in classification and regression problems. LR is applied when the dependent (target) variable is binary and makes no assumptions about the distribution of independent variables [26].

Ensemble learning is a technique that uses multiple models together. By utilizing several models, it produces more accurate and robust predictions. In other words, instead of a single model, predictions from various models are used to provide more correct results. This technique mitigates the shortcomings of each model while striving to deliver more robust and trustworthy predictions. It proves particularly valuable in addressing the challenges posed by complex and demanding datasets.

Ensemble learning employs a variety of algorithms and processes the features derived from the dataset to formulate predictions. These predictions are aggregated through various voting methods to create stronger and more dependable results, particularly when dealing with high-dimensional or unbalanced data [27]. In this research, we use manually selected/default parameter values without hyperparameter optimization to measure the baseline performance of the algorithms. The chosen hyperparameter values are in line with common usage in the literature and in the Scikit-learn library [28]. The XGBoost algorithm [29] and the Word2Vec model [24] were used within the range of values recommended in the reference article. In the LR model, which is one of the traditional machine learning techniques, since the ‘saga’ method is recommended as the solver parameter for large data sets, this parameter was set to balanced, the class_weight parameter was set to ‘balanced’ to preserve the data classification distribution, and the number of iterations was first set to 1500 and the algorithm was run. However, since the algorithm run with this iteration value was insufficient for the size of the dataset, the problem was solved by increasing the number of iterations from 1500 to 2000. In the SVM algorithm, the parameters consist of the default values in Scikit-learn. However, in this library, the probability value is False by default, whereas in our research we set the probability value to True to find out how confident the model is for the ROC-AUC evaluation metrics and to get the probability values for each prediction. The parameter details of the algorithms used in the study are shown in Table 2.

Table 2. Parameter values used in the algorithms

Model Name	Parameters
Word2Vec	vector_size=100, window=5, min_count=2, workers=4, epochs=10
LR	C=1.0, solver='saga', max_iter=2000, class_weight='balanced'
kNN	metric='euclidean', n_neighbors=2
SVM	C=1.0, kernel='rbf', probability=True
DT	default
NB	default
XGBoost	n_estimators=100, learning_rate=0.1, random_state=42

3.4. Evaluation Metrics

In order to measure the performance success of the algorithms, accuracy, precision, recall, F1-score metrics provided in (3.1)-(3.4) as well as Receiver Operating Characteristic (ROC) curve were used.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (3.1)$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3.2)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3.3)$$

$$\text{F1 - Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.4)$$

The ROC curve was constructed using the real class labels and the estimated probabilities. Using the J-Index method [30], the threshold value at which the maximum difference between the true positive and false positive rates was determined as the optimum threshold value. The optimal threshold value was used as a decision mechanism to determine whether the user was spam. This approach ensures that rating-based and review-based classifications are performed on the same data, aiming for more accurate and reliable results.

$$J = \text{True Positive Rate} - \text{False Positive Rate}$$

3.5. Experiments and Results

In the research, fake user detection was performed using rating data and product reviews with kNN, SVM, NB, LR, and DT machine learning algorithms. At the same time, the research aims to utilize the advantages of ensemble learning methods. For this reason, machine learning techniques were used with two different approaches, Stack and Voting Classifier. In the Voting Classifier method, majority voting was performed based on the outputs of the machine learning algorithms used in the study. The final predictions were selected according to the majority vote. In the Stack Classifier method, the algorithm's predictions were given to the XGBoost algorithm as input, and this ensemble learning algorithm decided the final results. Figure 1 outlines the general framework of the study.

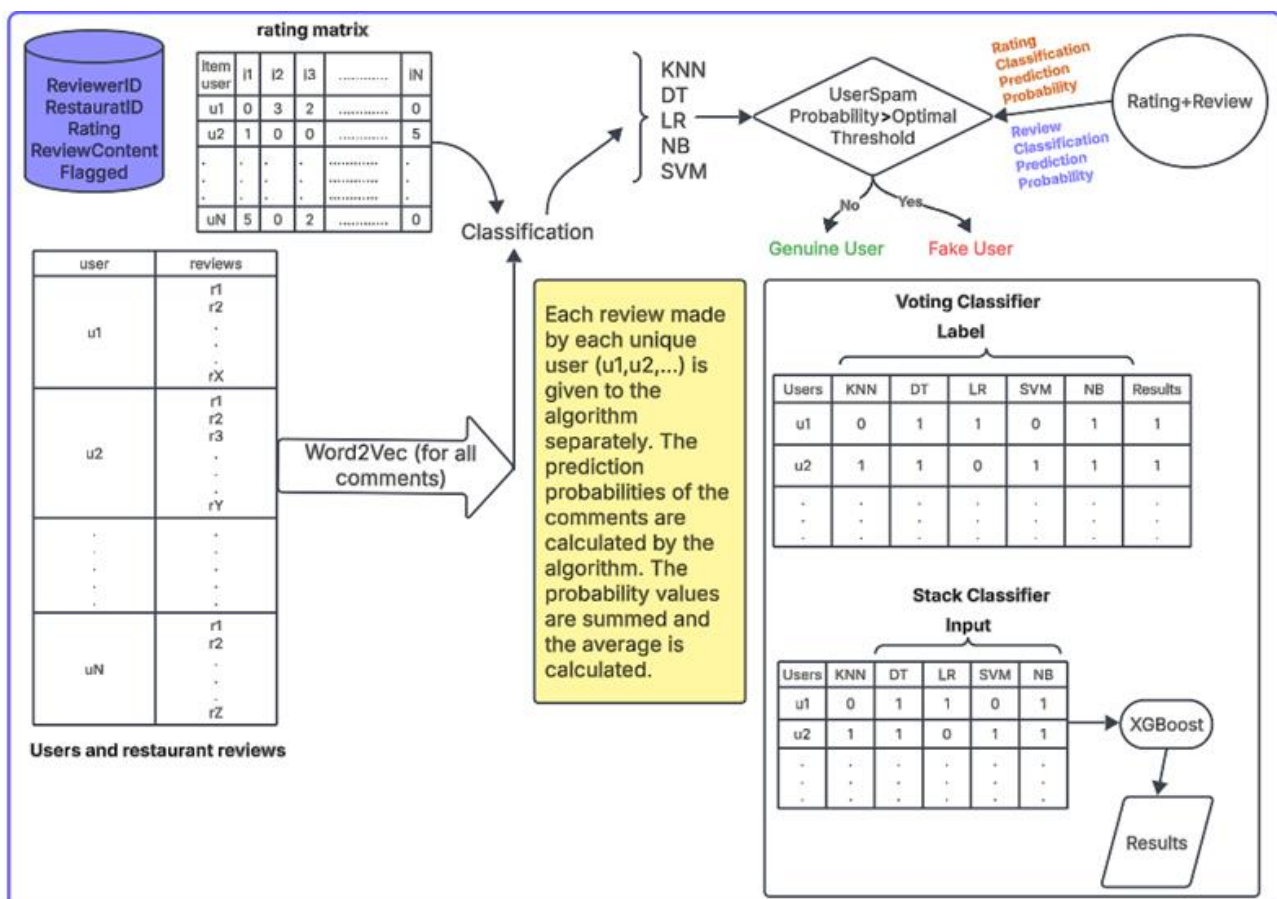


Figure 1. Process stages

Figure 2 shows the classification results based on rating data. When the curve is analyzed, it is seen that the model's performances are very close to each other. It can be said that classification success is not sufficient when only rating data is used.

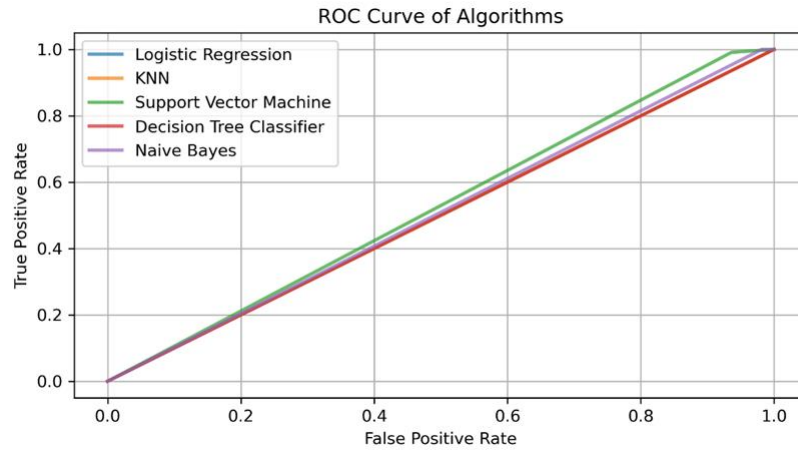


Figure 2. Classification performance of the algorithms using only rating data

Table 3, which presents the detailed classification results of the algorithms, shows that LR, kNN, and DT models have the same performance values with an F1 score of approximately 76.74%. The fact that these models have a recall of 100% indicates that they correctly classified all positive examples, but their precision is low. This suggests that the false positive predictions of the models are high. The NB algorithm outperforms these three models with an F1 score of 77.07%. The SVM algorithm has the highest F1 score of 77.47% among all algorithms. It can be concluded that the algorithms have an average performance in detecting fake users.

Table 3. Classification results using only rating data

Model Name	Accuracy	Precision	Recall	F1 Score
LR	62.26	62.26	100.0	76.74
kNN	62.26	62.26	100.0	76.74
SVM	64.12	63.59	99.1	77.47
DT	62.26	62.26	100.0	76.74
NB	62.95	62.69	100.0	77.07

Figure 3 shows the results of the review-based machine learning classification. Table 4 analyzes the classification results of all algorithms using product review data features against evaluation metrics. LR and SVM have the highest accuracy that are equal to 73.61% and 72.82%, respectively. The fact that kNN and LR models have the best recall value makes it possible to conclude that these algorithms are more successful in detecting fake users. The DT algorithm showed the lowest results. This method achieved 61.64% accuracy and 57.57% recall, suggesting the model may be overfitting. As a general analysis, it can be concluded that although the SVM algorithm has the highest precision, the LR model shows a more balanced success in all metrics.

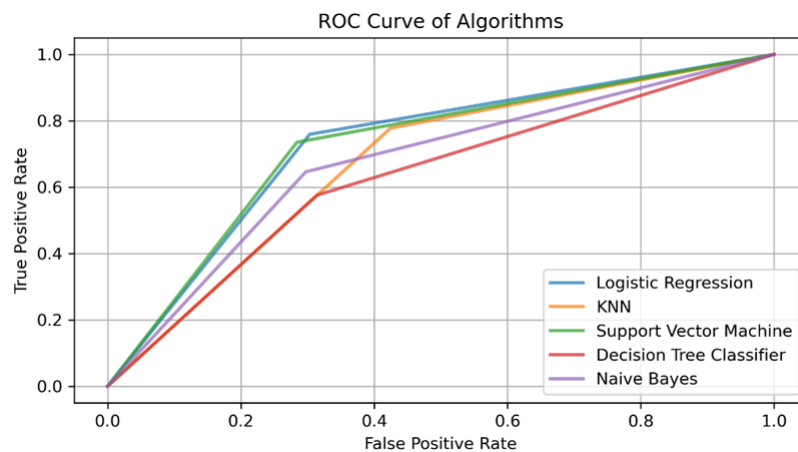
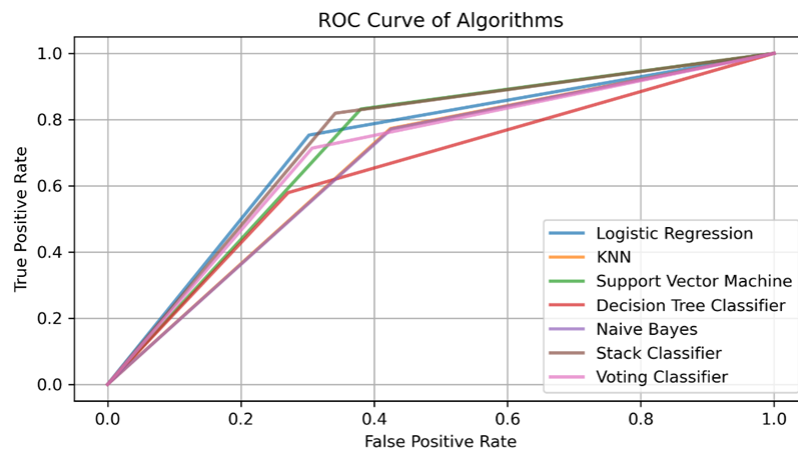


Figure 3. Classification performance of the algorithms using only user review data

Table 4. Classification results using product review data

Model Name	Accuracy	Precision	Recall	F1 Score
LR	73.61	81.01	75.93	78.39
kNN	70.29	75.74	77.76	76.74
SVM	72.82	81.51	73.54	77.32
DT	61.64	75.74	57.57	65.41
NB	66.70	78.73	64.60	70.97

As with the classification of review data, the prediction probabilities for rating data were given equal weighting values for the classification results obtained separately from rating and review data. The classification was performed separately for both data types. As a result of the classification, the prediction probability of the classification made according to both data types were averaged, and the final result was obtained. The obtained probability value was compared with the optimum threshold value (as in the case of review data classification only) to decide whether the user is real or fake. Finally, in Figure 4 and Table 5, we analyze and present our rating results and review data using the ensemble learning method, Stacking and Voting Classifier. The Stack Classifier model shows the best performance according to the ROC curve. Here, in the Stack Classifier model, machine learning techniques were used as the base model and the XGBoost algorithm as the meta-model. The prediction values of the machine learning techniques used herein were given as input values to the final classifier, the XGBoost algorithm. The VotingClassifier method takes the predictions of the machine learning algorithms used in the study and makes the final decision based on the prediction value labeled by the majority.

**Figure 4.** Classification performance of the algorithms when rating and review data are combined

When Table 5 is examined in detail, the highest accuracy rate is in the Stack Classifier model, with an F1 score of 81.11%. As a result of the LR and Stack Classifier models having the highest precision values of 80.94% and 80.33%, it can be said that the number of false positives is low. Thus, users can be prevented from being labeled as false users. Except for the DT, other algorithms also showed good and balanced performance. The NB algorithm showed relatively less performance.

Table 5. Classification results in machine learning and ensemble learning (using rating and review data)

Model Name	Accuracy	Precision	Recall	F1 Score
LR	73.26	80.94	75.30	78.02
kNN	69.98	75.62	77.27	76.44
SVM	75.34	78.85	83.18	80.96
DT	63.46	78.46	57.92	66.64
NB	69.84	75.53	77.13	76.32
Stack Classifier	75.96	80.33	81.91	81.11
Voting Classifier	70.60	79.84	71.36	75.36

Boldfaced values indicate the "best" performances.

It can be said that using the XGBoost algorithm as a meta-model contributes positively to the success of machine learning techniques. Using rating and review data in detecting fake users has positively affected the success. Only the kNN algorithm has shown relatively less success than the classification based on product reviews. The success has increased even more due to using machine learning algorithms with ensemble learning algorithms.

3.6. Performance Comparison with Previous Work

Supervised machine learning techniques have been used in previous work [15] to detect fake users on the YELP dataset. In their study, they used NB, SVM, LR algorithms to detect fake profiles on the review data. In the results of the study, the LR algorithm achieved 78%, the NB algorithm 65% and the SVM algorithm 77% F1- score success. In another study [22], fake user detection was performed on a different YELP dataset by combining product reviews with some behavioral characteristics of users (total number of capital letters, punctuation marks and emojis). Here, kNN, SVM, NB, LR algorithms achieved an F1-score of 82.40%, 82.17%, 81.86%, 82.20% respectively.

Different YELP datasets were used in both studies. In our research, we used the open-access YELP dataset available on kaggle.com, which is different from the above studies. Despite the differences in the dataset, fake user detection using only reviews achieved almost similar results. Although a direct comparison is not appropriate given the differences, it can be said that the proposed multi-data approach shows a significant and promising classification success on the used dataset.

4. Conclusion

In this study, the detection of fake users on digital platforms was carried out by combining the classification of two separate data sources: rating data and product reviews. Unlike traditional approaches, this method analyzes both the rating behaviors and textual comments of users, allowing for more effective identification of fake profiles. Two classifications were performed using machine learning algorithms to separate fake users from real ones. The dataset used in this study has sparse architecture. Although the dataset has much user information, the number of products these users evaluate is close to the minimum. This may cause less meaningful features to be extracted from users. However, the classification results showed that rating and review data overcame this disadvantage and increased detection success. In addition, the study combined machine learning techniques with ensemble learning techniques, and a second approach was proposed. The classification results obtained from this approach showed that ensemble learning methods further improved the classification success. The results of this research aim to increase the accuracy of recommendation systems, improve user satisfaction, and support the long-term success of the platforms. However, the small number of labeled data and unbalanced data distribution have partially limited the study. In future work, integrating additional data sources, such as behavioral features reflecting user and item interactions, can strengthen fake user detection. In this way, user behavior can be better analyzed, and more meaningful features can be extracted to improve classification success. In addition, using more advanced transformer models such as BERT, GPT and more advanced architectures for text data can significantly improve detection performance. Fake users are a minority class compared to real users. This leads to the problem of data imbalance. Using techniques to reduce class imbalance such as artificial data generation can improve the accuracy of the model. These enhancements could further strengthen the detection of fake users and improve the security and reliability of social platforms, especially e-commerce.

Author Contributions

All the authors equally contributed to this work. They all read and approved the final version of the paper.

Conflict of Interest

All the authors declare no conflict of interest.

Ethical Review and Approval

No approval from the Board of Ethics is required.

References

- [1] C. Lin, S. Chen, M. Zeng, S. Zhang, M. Gao, H. Li, *Shilling black-box recommender systems by learning to generate fake user profiles*, IEEE Transactions on Neural Networks and Learning Systems 35 (1) (2022) 1305-1319.
- [2] T. T. Nguyen, N. Quoc Viet Hung, T. T. Nguyen, T. T. Huynh, T. T. Nguyen, M. Weidlich, H. Yin, *Manipulating recommender systems: a survey of poisoning attacks and countermeasures*, ACM Computing Surveys 57 (1) (2024) 1-39.
- [3] J. Suryawanshi, S. M. Abdul, R. P. Lal, A. Aramanda, N. Hoque, N. Yusoff, *Enhanced recommender systems with the removal of fake user profiles*, Procedia Computer Science 235 (2024) 347-360.
- [4] R. A. Zayed, L. F. Ibrahim, H. A. Hefny, H. A. Salman, A. Almohimeed, *Using ensemble method to detect attacks in the recommender system*, IEEE Access 11 (2023) 111315-111323.
- [5] Q. Zhou, J. Wu, L. Duan, *Recommendation attack detection based on deep learning*, Journal of Information Security and Applications 52 (2020) 102493.
- [6] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, *Effective attack models for shilling item-based collaborative filtering system*, In Proceedings of the 2005 WebKDD Workshop, held in conjunction with ACM SIGKDD, Chicago, Illinois, 2005.
- [7] S. Shaw, A. Singh, *Using machine learning algorithms to alleviate the shilling attack in a recommendation system*, 11 (5) (2023) 1879-1883.
- [8] S. Rani, M. Kaur, M. Kumar, V. Ravi, U. Ghosh, J. R. Mohanty, *Detection of shilling attack in recommender system for YouTube video statistics using machine learning techniques*, Soft Computing 27 (1) (2023) 377-389.
- [9] J. Cao, Z. Wu, B. Mao, Y. Zhang, *Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system*, World Wide Web 16 (5-6) (2013) 729-748.
- [10] A. B. Chopra, V. S. Dixit, *An adaptive RNN algorithm to detect shilling attacks for online products in hybrid recommender system*, Journal of Intelligent Systems 31 (1) (2022) 1133-1149.
- [11] R. A. Duma, Z. Niu, A. S. Nyamawe, J. Tchaye-Kondi, A. A. Yusuf, *A Deep Hybrid Model for fake review detection by jointly leveraging review text, overall ratings, and aspect ratings*, Soft Computing 27 (10) (2023) 6281-6296.
- [12] Y. Cheng, J. Guo, S. Long, Y. Wu, M. Sun, R. Zhang, *Advanced Financial Fraud Detection Using GNN-CL Model*, 2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Ottawa, ON, Canada, 2024, pp. 453-460.
- [13] Z. Han, T. Zhou, G. Chen, J. Chen, C. Fu, *A Robust Rating Prediction Model for Recommendation Systems Based on Fake User Detection and Multi-Layer Feature Fusion*, Big Data Mining and Analytics 8 (2) (2025) 292-309.
- [14] S. Rayana, L. Akoglu, *Collective opinion spam detection: Bridging review networks and metadata*, in: L. Cao, C. Zhang (Eds.) KDD '15: The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW Australia, 2015, pp. 985-994.
- [15] A. Sihombing, A. C. M. Fong, *Fake Review Detection on Yelp Dataset Using Classification Techniques in Machine Learning*, Proceedings of the 4th International Conference on Contemporary Computing

- and Informatics, IC3I, Singapore, 2019, pp. 64–68.
- [16] R. Barbado, O. Araque, C. A. Iglesias, *A framework for fake review detection in online consumer electronics retailers*, Information Processing and Management 56 (4) (2019) 1234-1244.
 - [17] Y. Jian, X. Chen, X. Wang, Y. Liu, X. Chen, X. Lan, W. Wang, H. Wang, *A metadata-aware detection model for fake restaurant reviews based on multimodal fusion*, Neural Computing and Applications 37 (1) (2024) 475–498.
 - [18] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, *Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews*, Technical Report, Department of Computer Science (UIC-CS-2013-03) University of Illinois (2013) Chicago.
 - [19] W. Zhou, J. Wen, Q. Xiong, M. Gao, J. Zeng, *SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems*, Neurocomputing 210 (2016) 197–205.
 - [20] H. İ. Ayaz, Z. Kamışlı Öztürk, *Shilling attack detection with one class support vector machines*, Necmettin Erbakan University Journal of Science and Engineering 5 (2) (2023) 246-256.
 - [21] P. K. Singh, P. K. D. Pramanik, N. Sinhababu, P. Choudhury, *Detecting unknown shilling attacks in recommendation systems*, Wireless Personal Communications 137 (1) (2024) 259–286.
 - [22] A. M. Elmogy, U. Tariq, A. Ibrahim, A. Mohammed, *Fake Reviews Detection using Supervised Machine Learning*, IJACSA) International Journal of Advanced Computer Science and Applications 12 (1) 2021 601-606.
 - [23] A. Dheyaa, *YelpReviewsDB*, <https://www.kaggle.com/datasets>, Accessed 2 April 2025.
 - [24] T. Mikolov, K. Chen, G. Corrado, J. Dean, *Efficient estimation of word representations in vector space*, (2013), <https://arxiv.org/pdf/1301.3781>, Accessed 2 April 2025.
 - [25] B. Mahesh, *Machine learning algorithms - a review*, International Journal of Science and Research 9 (1) (2020) 381-386.
 - [26] I. H. Sarker, *Machine learning: algorithms, real-world applications and research directions*, SN Computer Science 2 (3) (2021) 160.
 - [27] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, *A survey on ensemble learning*, Frontiers of Computer Science 14 (2) (2020) 241-258.
 - [28] Scikit-learn, *scikit-learn: Machine Learning in Python*, <https://scikit-learn.org/stable>, Accessed 28 June 2025.
 - [29] T. Chen, C. Guestrin, *XGBoost: A scalable tree boosting system*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016, pp. 785-794.
 - [30] W. J. Youden, *Index for rating diagnostic tests*, Cancer 3 (1) (1950) 32-35.