International Journal of Safety and Security Engineering
Vol. 14, No. 4, August, 2024, pp. 1243-1251
Journal homepage: http://iieta.org/journals/ijsse

Knowledge Rules-Based Decision Tree Classifier Model for Effective Fake Accounts Detection in Social Networks

Susan Mohammed[1*], Nabeel Al-Aaraji[1], Ahmed Al-Saleh[2]

[1] Software Department, IT College, University of Babylon, Babel 51002, Iraq
[2] Information Networks Department, IT College, University of Babylon, Babel 51002, Iraq

Corresponding Author Email: susanmohammed@itnet.uobabylon.edu.iq

https://doi.org/10.18280/ijsse.140421

Received: 29 April 2024
Revised: 2 July 2024
Accepted: 17 July 2024
Available online: 30 August 2024

Keywords:
social network accounts, fake accounts detection, machine learning, knowledge rules extraction, decision tree classifier

ABSTRACT

Most social media sites host numerous fake accounts, causing significant harm such as sending fake messages, spreading false news, wasting money, damaging reputations, and creating legal issues. Due to the rapid growth in social media users, detecting these accounts is crucial. This study aims to develop an effective and accurate model for detecting fake accounts using advanced machine-learning algorithms. The proposed model, called the Knowledge Rules-based Decision Tree Classifier, leverages machine learning techniques to extract knowledge rules specifically designed for identifying fake accounts. The model employs a decision tree algorithm to derive these rules from input social network accounts and applies feature selection algorithms to minimize the number of extracted rules, thus enhancing detection efficiency. Tested on Twitter and Instagram, the model achieved 100% accuracy, demonstrating its effectiveness and reliability. The innovative aspect of this research lies in its novel use of machine learning-derived knowledge rules for fake account detection. This pioneering approach offers a robust solution to mitigate the significant harms associated with fake accounts on social media platforms, providing a reliable method to ensure the integrity of social network interactions.

1. INTRODUCTION

The widespread utilization of social media platforms has transformed communication by linking people worldwide and enabling the sharing of thoughts, knowledge, and personal encounters. Nevertheless, the increase in the number of fraudulent accounts on these platforms has emerged as a significant concern, endangering their genuineness, trustworthiness, and safety. These fake accounts, often known as bots, are created to spread misinformation, influence public perception, and execute harmful activities [1].

Researchers, politicians, and platform administrators are keen on identifying counterfeit profiles on social networks. Conventional methods for detecting fraudulent accounts mostly rely on manual investigation, which is time-consuming and frequently unable to keep up with the constantly evolving nature of fraudulent activities. In recent years, machine learning techniques have emerged as valuable tools for addressing this challenge by automating the process of detecting fraudulent accounts [2].

Machine learning algorithms analyze patterns, behavior, and characteristics to distinguish between genuine and counterfeit accounts accurately by leveraging extensive data gathered on these platforms [3].

By analyzing existing characteristics, specialists categorized social media accounts by using a range of detection methods. Employing machine learning techniques further enhances precision in categorizing these accounts and distinguishing between fraudulent and genuine ones [4].

Patil et al. [5] proposed a novel approach for detecting and classifying fake social media profiles using the majority voting technique. Their method combined multiple machine learning algorithms, such as Decision Trees, XGBoost, Random Forest, Extra Trees, Logistic Regression, AdaBoost, and K-Nearest Neighbors, each designed to capture different aspects of user behavior and profile characteristics. By integrating these diverse algorithms, they created an ensemble of classifiers, which were then subjected to a majority voting mechanism to determine the authenticity of a social media profile. Their results demonstrated that the Majority Voting Technique outperformed individual classifiers, achieving an accuracy, precision, recall, and F1-score of 99.12%.

Smruthi and Harini [6] discussed the results of utilizing Facebook's functionalities in detecting fake profiles. They evaluated the accuracy of various supervised machine learning methods, such as the *k*-nearest Neighbor (*k*-NN), support vector machine (SVM), Naive Bayes, decision tree, and random forest algorithms, by using selected features to distinguish between genuine and false accounts. Performance was assessed through supervised machine learning techniques, achieving an accuracy rate of 80%.

In their study, Amey Bhovar [7] employed various supervised classification algorithms, such as *k*-NN, decision tree, naive Bayes, random forest, and SVM, to categorize a

Twitter network. Their findings revealed that the decision tree algorithm exhibited the highest accuracy. This algorithm was selected as the "best" algorithm not only due to its superior accuracy but also because of its relatively straightforward and comprehensible nature compared with the other algorithms.

Voitovych et al. [8] used a support vector machine as the foundation for creating a decision-making system. Multiple experimental studies were conducted on Facebook, encompassing account analysis, parameter extraction, and parameter selection. Using a customized dataset that contained the characteristics of legitimate and fraudulent accounts, the classifier achieved an accuracy of 97% in identifying fraudulent accounts.

In their study, Kondeti et al. [9] applied SVM, *k*-NN, random forest, logistic algorithms, and z-score and Min-Max normalization techniques to predict fake users. These techniques increased accuracy to 98%.

Kumar et al. [10] classified and detected fraudulent accounts. They utilized machine learning algorithms, including random forest, SVM, and XG boost. Their study revealed that machine learning algorithms can effectively identify patterns and anomalies that are indicative of fake accounts with high accuracy. Considering additional factors, such as post metadata, account activity, and network parameters, significantly enhanced the accuracy of the results.

The primary objective of the current study is to develop a new machine-learning model for accurately and efficiently classifying fake accounts in social networks.

2. OVERVIEW OF FAKE ACCOUNTS

Social networks have become dominant forces nowadays, with the number of users on social media sites increasing year after year. The primary benefit of online social media is that people from all different parts of the world can quickly interact and communicate with one another. However, this widespread connectivity has also given rise to malicious activities, such as the proliferation of fake identities and spam. Surveys conducted by Twitter and Instagram indicate that the number of accounts created exceeds the actual number of genuine users on their platforms. This situation suggests a growing prevalence of fake profiles [11].

Creating fake accounts on social media serves various purposes, including:

·Generating hateful posts
·Online impersonation
·Advertising and campaigning
·Privacy intrusion

Differentiating fake accounts from genuine users poses a challenge because social media spammers frequently operate within legal boundaries. Moreover, fraudsters can utilize inexpensive automated techniques, and thus, detecting their deceptive practices is difficult for a large population of social media users. Hence, the identification and classification of social media accounts become crucial tasks, aiming to distinguish legitimate users from fake ones based on their unique characteristics. As an attribute, identity plays a pivotal role in setting individuals apart from one another [12].

Existing machine learning algorithms used for detecting fake identities frequently exhibit low accuracy and inefficiency. Thus, methods that offer higher accuracy and lower false positive ratios are necessary [13].

3. FAKE ACCOUNT DETECTION METHODS IN SOCIAL NETWORKS

In recent years, numerous studies have been conducted on fake account detection. Various parameters have been suggested for detecting fake accounts. Static parameters are typically utilized for ad hoc analysis, such as profile picture, name, date of birth, number of friends, photos, and likes. Some of these parameters are changing, such as the number of friends, and can be utilized for dynamic analysis that varies over time. Several techniques include the use of behavior criteria for the graph creation of social media links or the online analysis of account modifications. Various approaches for detecting fake accounts are provided based on their input parameters [12].

·Rule-based systems: These methods rely on predefined rules or heuristics to identify fake accounts. Rules may include criteria, such as suspicious behavior patterns, excessive posting frequency, or repetitive content. However, rule-based systems may have limitations in detecting sophisticated fake accounts that mimic genuine user behavior.

·Machine learning algorithms: Machine learning techniques have gained popularity in fake account detection. These methods involve training a model on labeled data, where features extracted from user profiles, network structure, or user activity are used to classify accounts as genuine or fake. Decision trees, neural networks, random forest, and SVM are the most common machine learning algorithms.

·Social network analysis: Social network analysis techniques are used to detect fake accounts by using the relationships and interactions among users. This method focuses on locating connection patterns, such as groups of fictitious accounts that behave similarly or carry out coordinated actions.

·Natural language processing (NLP): Textual content, such as user profiles, posts, comments, and messages, are analyzed using NLP algorithms to spot linguistic patterns connected to phony accounts. Sentiment evaluation, modeling of topics, and linguistic inconsistencies are examples of this.

·Image analysis: Image analysis algorithms can be used to identify phony accounts because profile images and other visual content are used more frequently. Image modifications, recognition of stock photos or stolen images, and assessment of the authenticity of profile pictures can be detected using these methods.

·Hybrid approaches: To improve detection accuracy, some methods are combined. For example, integrating machine learning algorithms into social network analysis or NLP approaches into picture analysis can provide a more comprehensive approach to detecting bogus accounts.

The effectiveness of these methods can vary depending on the quality and availability of data, the evolving techniques used by fake account creators, and the specific characteristics of the social media platform being analyzed [14].

4. MACHINE LEARNING TECHNIQUES FOR DETECTING FAKE ACCOUNTS IN SOCIAL NETWORKS

The number of fake accounts has increased significantly in the last few years, and thus, distinguishing them from actual accounts has become difficult. The most frequently used

techniques for detecting bots are supervised machine learning models [13].

Machine learning is a subfield of artificial intelligence that focuses on the development of algorithms and statistical models that enable computers to improve their performance on a specific task by learning from data, without being explicitly programmed. In essence, machine learning systems can automatically learn and adapt from experience. Various types of machine learning techniques are used to detect fake accounts. The most frequently used machine learning algorithms are decision tree and random forest. Although these algorithms may detect and identify bots and real accounts, they only produce satisfactory results. If a machine-learning model is improved with multiple machine-learning methods, then the total accuracy level can be increased. Many existing techniques detect false identification numbers by using simply a few attributes. The relevance of qualities influences decision-making accuracy. Consequently, accuracy will increase as the number of qualities increases [14].

Random forest algorithm

Random forest is a classifier that uses the average of several decision trees on different subsets of a given dataset to increase its performance prediction. It is used as a collection of forecasts from each tree of the decision trees to predict the final output based on the majority vote of predictions [15].

Decision tree algorithm

This algorithm is a popular machine-learning technique that uses a tree-like architecture to predict outcomes based on input features. The tree-like model is built by partitioning the feature space recursively into sections that correspond to various classes. At each node of the tree, partitions are selected by choosing the characteristic that provides the greatest information gain or reduction in entropy [16].

The current study used various indicators to evaluate the proposed model's effectiveness in detecting fake social network accounts, including a 2D matrix that represented the actual and predicted class. The confusion matrix described its composition by using true positives, false positives, true negatives, and false negatives. In addition, several performance metrics, such as accuracy, precision, recall, and F1 score, were applied.

A confusion matrix is a table that is designed to aid in the display of the various outcomes of a classification problem's forecast and results. It generates a table that contains all the predicted and actual values of a classifier. Four different combinations from the predicted and actual values of a classifier can be obtained, as shown in Figure 1 [17]:

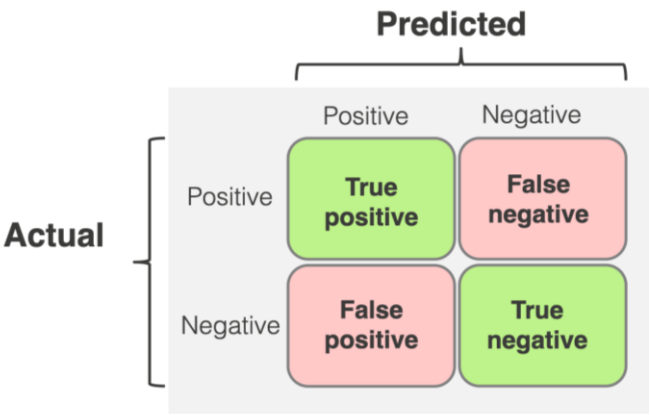

Figure 1. Confusion matrix

• ***True Positive:*** The number of times that the actual positive values match the predicted positive value.

• ***False Positive:*** The number of times that the model incorrectly predicts negative values as positive.

• ***True Negative:*** The frequency with which real negative values are equal to the expected negative values.

• ***False Negative:*** The number of times that the model incorrectly predicts positive values as negative.

Accuracy: Accuracy is used to compute the proportion of correctly categorized values. It is equal to the total number of values divided by the sum of all true values.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Precision: Precision is used to calculate the model's ability to categorize positive values accurately. It is derived by dividing the total number of expected positive values by the number of genuine positives.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$

Recall: Recall determines the model's ability to predict positive values. "How often does the model correctly predict positive values?" It is the number of genuine positives divided by the total number of positive values.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (3)$$

F1 Score: This score is the symbiotic relationship between recall and precision. It is useful when precision and recall must be considered.

$$\text{F1 Score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} * \text{Recall}) \quad (4)$$

A receiver operating characteristic (ROC) curve is an essential tool for assessing the performance of binary classification models. It visualizes the trade-off between the true positive rate (TPR) and the false positive rate (FPR) across different threshold values employed for classification decisions. In the context of machine learning, threshold values are used to determine whether a model's output score or probability should result in a positive or negative classification. The ROC curve is generated by plotting TPR against FPR at various thresholds, illustrating how model performance varies as these thresholds change. This visual representation allows analysts and data scientists to explore the sensitivity and specificity of a model across a range of decision criteria. A perfect classifier will exhibit an ROC curve that reaches the top-left corner, indicating a TPR of 1 and an FPR of 0, while a random classifier will follow a diagonal line where the TPR is equal to the FPR. Therefore, the ROC curve aids in making informed decisions about threshold selection and understanding the inherent trade-offs between true positives and false positives in binary classification scenarios [8].

5. PROPOSED KNOWLEDGE RULES-BASED DECISION TREE CLASSIFIER (KRDTC) MODEL

This study uses two social network datasets. The first dataset is the Fake Project dataset, which was released by the Italian National Research Council's (CNR) Institute of Informatics and Telematics. This dataset contains two types of

Twitter accounts: phony followers and actual accounts. It includes 11,737 Twitter accounts with a total of 12,030,893 tweets [18].

The second dataset is an Instagram dataset that consists of two types of accounts: fake and real accounts. This dataset contains 785 fake and real Instagram accounts with 785 posts. It consists of 692 fake accounts and 93 real accounts [19].

The proposed model, i.e., KRDTC, uses a decision tree classification algorithm to detect fake accounts effectively on social networks. It identifies and recognizes fake accounts on Twitter and Instagram datasets based on user attributes. It is composed of the following stages: data preprocessing, feature selection, classification (decision tree classifier), rule extraction, and model evaluation as demonstrated by Figure 2.

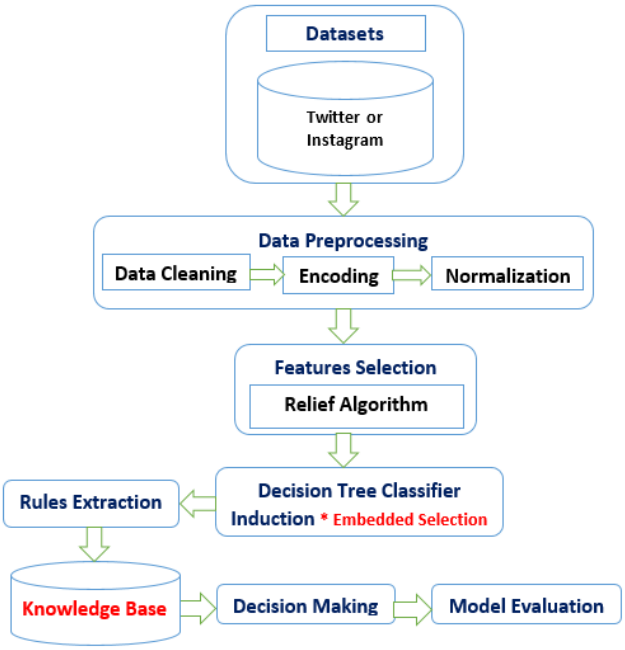

Figure 2. The proposed model architecture of KRDTC

5.1 Data preprocessing

Before applying the decision tree algorithm, the input social network dataset passes through many data preprocessing operations. The first preprocessing operation is removing noise and outliers. Rows and columns with missing values (empty rows or columns) are removed. The missing values included within a column are processed by replacing them with the mean value of that column. The second preprocessing operation is encoding. This process is critical because many machine learning algorithms operate with numerical data and may be incapable of dealing directly with categorical variables. The current study uses the label encoder method for encoding categorical labels into numerical values, which is frequently necessary when working with machine learning algorithms that require numerical input, such as the decision tree algorithm. The label encoder method is applied to each categorical column in the input dataset. The last preprocessing operation is normalization, which is performed to ensure that each feature has the same weight in the classification. In this research, z-score normalization is used as the technique to normalize each column within the input datasets.

The z-score normalization approach scales a feature's values to have a mean of 0 and a standard deviation of 1. This step is accomplished by subtracting the feature's mean from each value and then dividing it by the standard deviation [20].

5.2 Feature selection

The next stage in the proposed model is feature selection. The purpose of feature selection is to choose the most informative features for training the model. The relief algorithm is adopted for feature selection. The feature selection process is only applied to the Twitter dataset but not to the Instagram dataset because the number of Twitter dataset features is more than 34 while that of the Instagram dataset is only 12. The number of Twitter dataset features reaches 13 after applying the relief algorithm.

The process of selecting useful features and rejecting unnecessary ones can be defined broadly as feature selection. The relief algorithm is a feature selection approach that weighs attributes by using the nearest neighbor [21].

5.3 Data splitting

Thereafter, each dataset was divided into training and testing sets by using a stratified sampling technique to ensure that the distribution of classes was consistent across all sets. The training set was used to train the decision tree classifier and tune the hyperparameters.

To generate the knowledge rules, the decision tree classification algorithm was implemented on the training set, which comprised 80%. The remaining 20% of the dataset was used for testing.

5.4 Rule extraction

The most important stage of the proposed KRDTC model is rule generation. It includes extracting knowledge rules from the decision tree classifier after training the classifier. The extraction of rules from a decision tree algorithm is a pivotal step in making complex machine learning models comprehensible and interpretable. Although powerful for classification tasks, decision trees are frequently perceived as "black boxes" due to their intricate branching structures. Rule extraction involves distilling these structures into human-readable "IF-THEN" statements, shedding light onto the decision-making process. This process typically begins at the tree's leaves and proceeds upward through recursive analysis. Each branch in the tree corresponds to a rule that encapsulates a set of conditions that lead to a specific outcome. These rules provide an intuitive understanding of the factors that influence predictions, and thus, interpreting why a particular decision was made becomes possible. Rule extraction not only enhances model transparency but also facilitates domain-specific insights, empowering users to apply model-generated knowledge to practical contexts, such as the classification of social network accounts into fake and real profiles. This capacity for rule extraction fosters trust and the adoption of machine learning solutions in critical applications where interpretability and accountability are paramount.

The number of generated knowledge rules differs from one dataset to another. It depends on the complication and distribution of data. The number of knowledge rules generated using the proposed model was only 3 for the Twitter dataset and it took an extremely short time. Thus, these knowledge rules can be used to detect fake accounts in real-time. The Instagram dataset needed nearly 130 knowledge rules to detect fake accounts. This difference in the number of generated

knowledge rules between the two datasets can be attributed to the complication and distribution of data as mentioned earlier. To our knowledge, this study is the first research that uses the knowledge rules generated from a classification model to classify dataset accounts.

The specific process of extracting knowledge rules from a decision tree classifier involves several steps:

1. Training the decision tree classifier:

The first step is to train the decision tree classifier using the given dataset. The dataset comprises various features that describe the social network accounts, with the target variable indicating whether an account is real or fake. During training, the decision tree algorithm recursively splits the data into subsets based on feature values, aiming to maximize the separation of classes (real vs. fake) at each node.

2. Generating the decision tree structure:

After training, the decision tree structure is established, consisting of nodes (decision points) and branches (paths) that lead to leaf nodes (final decisions). Each path from the root to a leaf node represents a set of conditions that determine the classification of an account.

3. Extracting IF-THEN rules:

Rule extraction begins at the leaf nodes, where each path from the root to a leaf node is traced to form a rule. An IF-THEN statement is created for each path, where the IF part lists the conditions (feature thresholds) along the path, and the THEN part specifies the classification outcome (real or fake).

For example, a rule might state: "IF Feature1 > Threshold1 AND Feature2 < Threshold2 THEN Account = Fake."

4. Recursive analysis and simplification:

The extracted rules are analyzed recursively, simplifying them by removing redundant conditions and merging similar rules where possible.

This step ensures that the rules are concise, easy to interpret, and cover the decision tree's decision-making logic effectively.

5. Validating and optimizing rules:

The validity of the extracted rules is verified by applying them to a validation dataset. This step ensures that the rules accurately classify the accounts as real or fake.

If necessary, further optimization is performed to improve the rules' accuracy and efficiency.

6. Implementation and application:

The final set of knowledge rules is implemented into the detection system. These rules can now be applied in real time to classify social network accounts based on the specified conditions.

The rules provide transparency and interpretability, allowing users to understand the basis of each classification decision.

Algorithm 1 presents the proposed KRDTC algorithm that is used to generate knowledge rules.

Algorithm 1: Proposed KRDTC algorithm

Input: *Dataset with features and target variable.*

Output*: knowledge rules in an if-then format*

1 *Begin*
2 *Start with the entire dataset as the root node.*
3 *Choose the best feature to divide the dataset based on a criterion.*
4 *Divide the dataset into subsets based on the chosen feature.*
5 *Repeat the process recursively for each subset, choosing the best feature at each level.*
6 *Stop when a predefined criterion is met.*
7 *Create leaf nodes with predicted class labels when the stopping criteria are met.*
8 *Traverse the tree to predict the class for new instances.*
9 *Adjust the tree structure during training to fit the data.*
10 *End*

The number of features present in the knowledge rules generated by applying the algorithm is fewer than the total number of features in the input dataset. This discrepancy can be attributed to the inherent feature selection mechanism within the decision tree algorithm. The algorithm inherently performs embedded feature selection when determining the best feature at each level to divide the dataset. Consequently, the decision tree algorithm naturally focuses on the most informative features, leading to a subset of selected features in the knowledge rules.

6. RESULTS AND DISCUSSION

To test the model, two datasets (from Twitter and Instagram) were selected in this study. These datasets were preprocessed using several steps. First, empty or missing values were removed to clean the missing data. Second, categorical features were encoded using the label encoder technique. Lastly, numerical features were normalized using the z-score normalization technique. Thereafter, the decision tree algorithm was used as a machine learning model. Every dataset was divided into 80% for training and 20% for testing the decision tree algorithm. The results showed that the confusion matrix, accuracy, precision, recall, and F1 score of the decision tree classification algorithm on the training datasets were used to quantify the performance of the proposed model to demonstrate how well it was able to distinguish between fake and real accounts. Thereafter, knowledge rules were extracted from the decision tree algorithm, tested with the testing data, and then evaluated using several performance metrics.

By applying the knowledge rules generated using the proposed KRDTC model, the performance metrics reached 100%, as shown in Figure 3, which displays the confusion matrix results of the Twitter dataset.

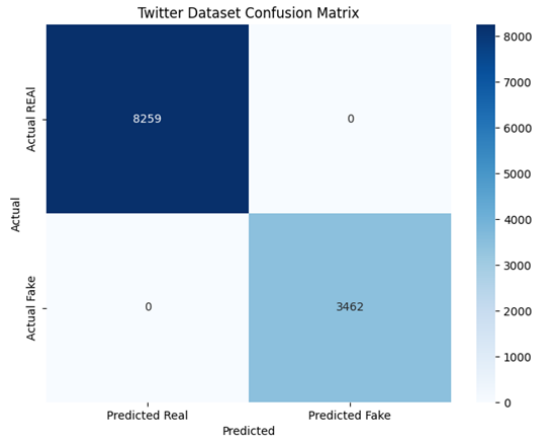

Figure 3. Twitter dataset confusion matrix

The confusion matrix of the Instagram dataset is shown in Figure 4.

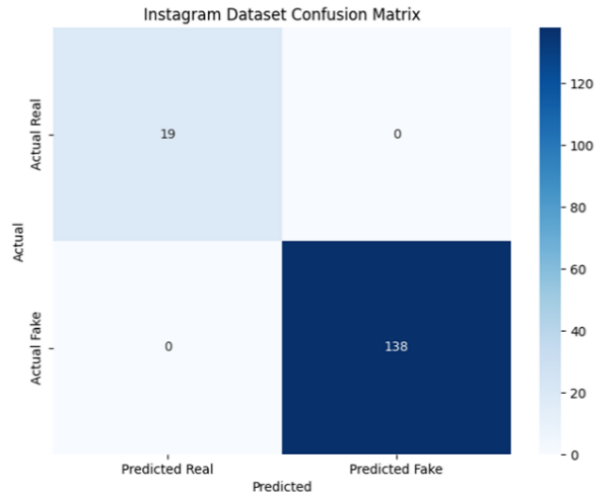

Figure 4. Instagram dataset confusion matrix

The measure used to evaluate the performance of the proposed classifier is the ROC curve. The ROC curve was generated for each one of the input datasets based on TPR and FPR.

The ROC results are shown in Figures 5 and 6 for the Twitter and Instagram datasets, respectively.

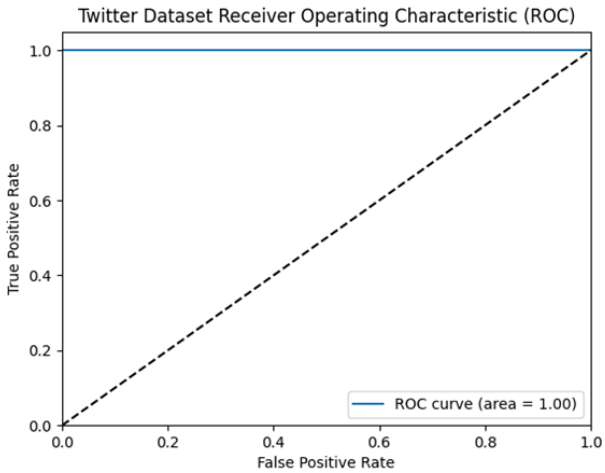

Figure 5. Twitter dataset ROC curve

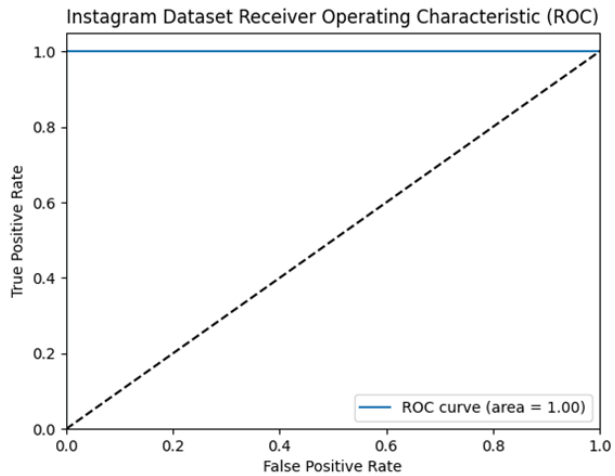

Figure 6. Instagram dataset ROC curve

7. ADVANTAGES OF THE PROPOSED MODEL

The KRDTC model demonstrates several significant advantages in the realm of fake account detection:

1. High accuracy: The model achieved 100% accuracy, precision, recall, and F1 score on both Twitter and Instagram datasets, showcasing its exceptional ability to correctly identify fake accounts.

2. Efficiency in rule extraction: By extracting knowledge rules from the decision tree, the model provides a transparent and interpretable set of criteria for classification. This not only enhances trust in the model's decisions but also allows for real-time detection of fake accounts.

3. Simplicity and speed: The decision tree algorithm's inherent simplicity and speed in generating rules reduce computational overhead, making the model highly efficient. This efficiency is crucial for handling large volumes of social media data in real-time applications.

4. Versatility: The model's success with different social networks (Twitter and Instagram) suggests its potential applicability to other platforms, such as Facebook and LinkedIn. This versatility makes the KRDTC model a robust solution for a wide range of online environments.

5. Reduction in complexity: The approach bypasses the need for extensive feature engineering and complex computations required by traditional methods. By focusing on decision tree-based rule extraction, the model simplifies the classification process, making it more accessible and practical for various applications.

6. Enhanced detection capabilities: The use of decision tree-generated rules ensures that the model captures nuanced patterns in the data, leading to more accurate and reliable detection of fake accounts. This capability is crucial in maintaining the integrity of social networks and protecting users from fraudulent activities.

By addressing these advantages, the proposed KRDTC model provides a comprehensive and efficient solution for fake account detection, offering significant improvements over existing methods in terms of accuracy, efficiency, and applicability across different social media platforms.

8. EVALUATION OF RESULTS

Performance measurements are necessary to evaluate the proposed model's effectiveness in detecting fake accounts on the input datasets. Performance metrics provide information regarding the performance of models or processes being evaluated. This study used various indicators to evaluate the proposed model's effectiveness, such as accuracy, precision, recall, and F1 score, which were applied to each dataset.

The performance metrics of the Twitter dataset are as follows:

·Accuracy: 100%,
·Precision: 100%,
·Recall: 100%, and
·F1 score: 100%.

Meanwhile, the performance metrics of the Instagram dataset are as follows:

·Accuracy: 100%,
·Precision: 100%,
·Recall: 100%, and
·F1 score: 100%.

The rules generated by the proposed KRDTC model for the

Twitter dataset are:

If (df1['crawled_at'] ≤ 1146.50) & (df1['updated'] > 31.50), then class = fake.
If (df1['crawled_at'] > 1146.50), then class = real.

Under the previous decision rules, several important pieces of information can be extracted on how the decision tree algorithm is making classification decisions for the CNR Twitter dataset, including the following:

1. Conditions: The conditions used for classification are based on the features of the CNR Twitter accounts. The conditions are related to the "crawled_at" and "updated" attributes.

2. Splitting conditions: The first rule uses two conditions to divide the data: "crawled_at" less than or equal to 1146.50 and "updated" more than 31.50. That is, the algorithm is segmenting accounts based on when they were crawled and how recently they were updated.

3. Leaf nodes: Leaf nodes have two types: "fake" and "real." They are the final classification labels assigned by the decision tree to accounts that satisfy the conditions in the respective rules.

4. Hierarchy: The first rule is evaluated before the second rule. If an account meets the conditions of the first rule, then it is classified as "fake." If it does not meet the conditions of the first rule, then the algorithm proceeds to the second rule and classifies the account as "real."

5. Threshold values: Values, such as 1146.50 and 31.50, are thresholds used to divide the data. Accounts with "crawled_at" values less than or equal to 1146.50 are evaluated against the additional "updated" condition. This situation implies that accounts crawled earlier than this threshold are being evaluated for their update frequency.

6. Interpretation: The first rule can be interpreted as suggesting that accounts crawled before a certain time (1146.50) but updated recently (greater than 31.50) are more likely to be classified as "fake." Conversely, the second rule suggests that accounts crawled after 1146.50 are likely to be classified as "real." This condition might indicate a pattern wherein recently updated accounts are considered more suspicious if they were created earlier.

7. Simplicity: The provided rules are relatively straightforward to understand. Thus, they are interpretable for a technical and nontechnical audience. This characteristic is one of the advantages of decision trees.

8. Feature importance: The provided rules mention only two features ("crawled_at" and "updated"). That is, the three features are the most important features of the input CNR datasets.

Some of the rules generated by the proposed KRDTC model for the Instagram dataset are as follows:

```
if username_has_number <= 0.73:
    if is_joined_recently <= 0.29:
        if edge_follow <= -0.60:
            if full_name_length <= -0.88:
                if is_private <= 0.81:
                    return 'FAKE_PROFILE'
                else:
                    if edge_follow <= -0.69:
                        return 'REAL_PROFILE'
                    else:
                        return 'FAKE_PROFILE'.
```

Under the previous decision rules, several important pieces of information can be extracted on how the decision tree algorithm is making classification decisions for the Instagram dataset, including the following.

1. Conditions: The conditions used for classification are based on the features of Instagram accounts. The conditions are related to many attributes.

2. Splitting conditions: Different rules use many conditions to divide the data.

3. Leaf nodes: Leaf nodes have two types: "fake" and "real". They are the final classification labels assigned by the decision tree to accounts that satisfy the conditions in the respective rules.

4. Hierarchy: The rules of the Instagram dataset are highly overlapping. If an account meets the conditions of the first rule, then it is classified as "fake." If it does not meet the conditions of the first rule, then the algorithm proceeds to the second rule and continues in this manner until it reaches the last rule, which ends with a leaf node that classifies the account as "real."

5. Threshold values: The Instagram dataset uses many threshold values to determine if an account is fake or real because many rules exist with various important features.

6. Simplicity: The provided rules are relatively straightforward, making them interpretable for technical and non-technical audiences. This feature is one of the advantages of decision trees.

The difference in the number of rules generated by the KRDTC model depends on the following reasons:

1. Complexity of the input dataset; and
2. Type, number, and importance of features that the dataset consists of.

Table 1 presents a comparison of the proposed KRDTC with previous related work models. As shown in the table, many machine-learning algorithms have been used to build classification models. The highest accuracy was 98% on the Twitter dataset. The proposed KRDTC model achieved the highest accuracy of 100%.

Table 1. Comparison of related studies

Reference	Year	Dataset	Algorithm	Accuracy
	2023	Twitter and Instagram	Proposed KRDTC Model	100%
[9]	2023	Instagram and Twitter	Linear Regression, Random Forest, Decision Tree, SVM, Naïve Bayes, XG Boost, *k*-NN, and Multilayer Perceptron (MLP)	95%
[6]	2022	Twitter	SVM, Naive Bayes, *k*-NN, Decision Tree, and Random Forest	-
[7]	2022	Facebook	SVM	97%
[8]	2021	Twitter	Logistic Algorithms, SVM, *k*-NN, Random Forest, with Min-Max Normalization and Z-Score Techniques	98%
[5]	2019	Instagram, Facebook, and Linked-in	SVM, Decision Tree, Random Forest, *k*-NN Algorithm, and Naïve Bayes' Algorithm	60-80%

9. CONCLUSIONS

1. The proposed model achieved exceptional results in predicting real and fake profiles, attaining 100% accuracy, F1 score, recall, and precision for the Twitter dataset with 3 rules, followed by a similar 100% performance on the Instagram dataset using 130 rules.

2. The model leveraged knowledge rules generated using the KRDTC algorithm, which significantly improved accuracy, achieving a perfect 100% accuracy rate.

3. By applying the rules generated from the KRDTC model, the study successfully identified fake accounts across different social networks, particularly the Twitter and Instagram datasets, indicating its potential applicability to other platforms, such as Facebook and LinkedIn.

4. The utilization of rules generated from the decision tree algorithm streamlined the classification process, resulting in a substantial reduction in the computational time required for identifying fraudulent profiles.

5. This approach bypassed the need for extensive feature extraction and complex computation, as frequently required by traditional methods for detecting fake accounts.

6. The inherent simplicity and speed of decision tree-based rules were harnessed, leading to a more efficient and time-saving classification.

7. In addition to maintaining high accuracy in fake account detection, the proposed method demonstrated remarkable efficiency in processing time.

8. These findings suggest that the approach holds promise as a solution for addressing the challenges associated with online identity verification, particularly in the context of the continuously evolving landscape of online social networks.

The innovation of this study lies in the novel application of knowledge rule extraction for fake account detection. Traditional methods typically require extensive feature engineering and complex computations. In contrast, our approach uses the decision tree classifier to generate human-readable IF-THEN rules, providing both transparency and interpretability in the classification process. These rules simplify the detection process and offer clear insights into the decision-making criteria, essential for understanding and explaining the results.

This method's ability to condense complex decision trees into a concise set of rules enables real-time detection of fake accounts with minimal computational overhead. By leveraging the strengths of decision tree algorithms, such as simplicity and speed, the approach becomes a practical and effective solution for real-world applications. Additionally, the versatility of this method across different social networks highlights its potential for broad adoption on various online platforms, setting a new standard in the field of fake account detection.

REFERENCES

[1] Ellaky, Z., Benabbou, F., Ouahabi, S. (2023). Systematic literature review of social media bots detection systems. Journal of King Saud University-Computer and Information Sciences, 35(5): 101551. https://doi.org/10.1016/j.jksuci.2023.04.004

[2] Martín-Gutiérrez, D., Hernández-Peñaloza, G., Hernández, A.B., Lozano-Diez, A., Álvarez, F. (2021). A deep learning approach for robust detection of bots in twitter using transformers. IEEE Access, 9: 54591-54601. https://doi.org/10.1109/ACCESS.2021.3068659

[3] Belfin, R.V., Grace Mary Kanaga, E., Kundu, S. (2020). Application of machine learning in the social network. In: De, S., Dey, S., Bhattacharyya, S. (eds) Recent Advances in Hybrid Metaheuristics for Data Clustering. John Wiley & Sons, Hoboken, New Jersey, pp. 61-83. https://doi.org/10.1002/9781119551621.ch4

[4] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., Alomari, D.M. (2023). Machine learning-based social media bot detection: A comprehensive literature review. Social Network Analysis and Mining, 13(1): 20. https://doi.org/10.1007/s13278-022-01020-5

[5] Patil, D.R., Pattewar, T.M., Punjabi, V.D., Pardeshi, S.M. (2024). Detecting fake social media profiles using the majority voting approach. EAI Endorsed Transactions on Scalable Information Systems, 11(3): 1-18. https://doi.org/10.4108/eetsis.4264

[6] Smruthi, M., Harini, N. (2019). A hybrid scheme for detecting fake accounts in facebook. International Journal of Recent Technology and Engineering (IJRTE), 7(5S3): 2277-3878.

[7] Amey Bhovar, S. (2023). A study of different methodologies to detect fake account on social media using machine learning. International Journal of Science and Research, 12(2): 53-57. https://doi.org/10.21275/SR23109103538

[8] Voitovych, O., Kupershtein, L., Holovenko, V. (2022). Detection of fake accounts in social networks. Cyber Security: Education, Science, Technology, 2(18): 86-98. https://doi.org/10.28925/2663-4023.2022.18.8698

[9] Kondeti, P., Yerramreddy, L.P., Pradhan, A., Swain, G. (2021). Fake account detection using machine learning. In: Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020, Springer, Singapore, pp. 791-802. https://doi.org/10.1007/978-981-15-5258-8_73

[10] Kumar, P.V., Vardhan, S.S., Kavya, Y., Singh, K.B. (2023). Fake accounts detection on social media (Instagram And Twitter). International Journal of Research in Engineering and Science, 11(3): 492-499.

[11] Khushboo Saraswat, D.N.T. (2020). A review on fake account detection in social media. International Journal for Research in Applied Science & Engineering Technology, 8(XII): 1002-1006. https://doi.org/10.22214/ijraset.2020.32627

[12] Ali, I., Ayub, M.N.B., Shivakumara, P., Noor, N.F.B.M. (2022). Fake news detection techniques on social media: A survey. Wireless Communications and Mobile Computing, 2022(1): 6072084. https://doi.org/10.1155/2022/6072084

[13] Suganya, R., Muthulakshmi, S., Venmuhilan, B., Kumar, K.V., Vignesh, G. (2021). Detect fake identities using improved machine learning algorithm. Journal of Physics: Conference Series, 1916: 012056. https://doi.org/10.1088/1742-6596/1916/1/012056

[14] Roy, P.K., Chahar, S. (2020). Fake profile detection on social networking websites: A comprehensive review. IEEE Transactions on Artificial Intelligence, 1(3): 271-285. https://doi.org/10.1109/TAI.2021.3064901

[15] Tunç, Ü., Atalar, E., Gargı, M.S., Aydın, Z.E. (2022). Classification of fake, bot, and real accounts on instagram using machine learning. Politeknik Dergisi, 27(2): 479-488.

https://doi.org/10.2339/politeknik.1136226
[16] Saranya Shree, S., Subhiksha, C., Subhashini, R. (2021). Prediction of fake Instagram profiles using machine learning. Available at SSRN 3802584. https://doi.org/10.2139/ssrn.3802584
[17] Murugesan, S., Pachamuthu, K. (2022). Fake news detection in the medical field using machine learning techniques. International Journal of Safety and Security Engineering, 12(6): 723-727. https://doi.org/10.18280/ijsse.120608
[18] Kodati, S., Reddy, K.P., Mekala, S., Murthy, P.S., Reddy, P.C.S. (2021). Detection of fake profiles on twitter using hybrid SVM algorithm. In E3S Web of Conferences, 309: 01046. https://doi.org/10.1051/e3sconf/202130901046
[19] Nahm, F.S. (2022). Receiver operating characteristic curve: Overview and practical use for clinicians. Korean Journal of Anesthesiology, 75(1): 25-36. https://doi.org/10.4097/kja.21209
[20] Akyon, F.C., Kalfaoglu, M.E. (2019). Instagram fake and automated account detection. In 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, pp. 1-7. https://doi.org/10.1109/ASYU48272.2019.8946437
[21] Urbanowicz, R.J., Meeker, M., La Cava, W., Olson, R. S., Moore, J.H. (2018). Relief-based feature selection: Introduction and review. Journal of Biomedical Informatics, 85: 189-203. https://doi.org/10.1016/j.jbi.2018.07.014