Explainable hybrid ensemble for fake account detection using DSM-FS and SMOTEENN

Discover Artificial Intelligence

https://doi.org/10.1007/s44163-026-01718-x

Article in Press

Explainable hybrid ensemble for fake account detection using DSM-FS and SMOTEENN

Received: 2 March 2026

Accepted: 24 June 2026

Published online: 07 July 2026

Cite this article as: Shamki Z.A.M., Alrammahi A.A.H., Sari F.A.O. *et al.* Explainable hybrid ensemble for fake account detection using DSM-FS and SMOTEENN. *Discov Artif Intell* (2026). https://doi.org/10.1007/s44163-026-01718-x

Zahraa Azhar Muhammad Shamki, Ali Abdulkarem Habib Alrammahi, Farah Abbas Obaid Sari, Mustafa Noaman Kadhim, Dhiah Al-Shammary, Ayman Ibaida, Assefa K. Teshome & Khandakar Ahmed

Explainable Hybrid Ensemble for Fake Account Detection Using DSM-FS and SMOTEENN

Zahraa Azhar Muhammad Shamki[1], Zahraa.alramahi@uokufa.edu.iq
Ali Abdulkarem Habib Alrammahi[1], alia.alramahi@uokufa.edu.iq
Farah Abbas Obaid Sari[1], faraha.altaee@uokufa.edu.iq
Mustafa Noaman Kadhim[2], mustafa.noaman@qu.edu.iq
Dhiah Al-Shammary[2], d.alshammary@qu.edu.iq
Ayman Ibaida[3], (Corresponding Author), ayman.ibaida@vu.edu.au
Assefa K. Teshome[3], Assefa.Teshome@vu.edu.au
Khandakar Ahmed[3], khandakar.ahmed@vu.edu.au
[1] *University of Kufa, Faculty of Computer Sciences and Mathematics, Najaf, Iraq.*
[2] *College of Computer Science and Information Technology, University of Al-Qadisiyah, Dewaniyah, Iraq.*
[3] *Intelligent Technology Innovation Lab, Victoria University, Melbourne, Victoria, Australia.*

Abstract: This paper introduces a method for fake accounts detection on social media through machine learning using several classification algorithms and sophisticated feature extraction. Preprocessing steps included dataset integration, feature normalization, entropy-based feature extraction, behavioral ratio computation, and class imbalance handling using SMOTE and SMOTEENN. Ensemble learning models, including LightGBM, CatBoost, XGBoost, and Logistic Regression, were trained on the training dataset and subsequently evaluated on the unseen testing dataset to assess their classification performance. Models were evaluated using accuracy, precision, recall, F1 score, and confusion matrices. SHAP explanation displayed the feature contributions. Experimental results demonstrated the effectiveness of the proposed approach, achieving the highest predictive accuracy of 99.71% and a balanced F1-score of 99.70%. These findings demonstrate excellent discriminative performance in distinguishing real accounts from the fake accounts and generated the following implications. Mixtures of feature selection, resampling strategies, and explainable AI can clearly enhance fake account detection. The proposed framework provides a scalable and effective solution for identifying fraudulent social media accounts and enhancing trust and security across online platforms.

Keywords: *Fake account detection dataset, Instagram fake profiles dataset, Machine learning algorithms, Ensemble model, Feature selection model and Explainable AI.*

1.Introduction

The growth of social media platform usage, especially that of Instagram, has created an online sphere with billions of real active accounts. However, a large number of fake or automated accounts have also emerged, used to either post spam, fake news, fraud or manipulate the opinion of society[1]. In this case, such accounts are used to increase social media metrics of a particular account, which pays for these actions. They may also be termed as fake followers. The most critical distinction between an automated account and a fake one is that the former boosts its own metrics, while the latter boosts the metrics of other users and therefore the discussion on social media becomes unfair [2]. Fake accounts are on the rise, and they need to be detected and removed. As a result, this study delves into fake Instagram accounts based on their profiles.

Detection and classification are critical aspects of social media network safety, security, and trustworthiness [3]. Machine learning has proven to be highly effective at spotting fake accounts based on user profile features, the model's behavior, and its interaction data [4]. For this problem, different classification models, decision tree-based algorithms, gradient boosting methods, and deep learning solutions have been tried. Each model has its benefits and drawbacks in terms of accuracy, generalization and computational cost [5]. This study focuses on the comparative assessment of classification techniques suitable for differentiating between authentic and fraudulent accounts on Instagram. We

establish this by training various models, ranging on efficacy and reliability over a wide range of configurations from clustering methods to tuning the threshold and verifying accounts in an automated way against real users that utilize the service. These results enable to learn in practice how the number of fake accounts can be reduced, and safely improve throughout the entire platform and thereby continuously ensure a high-quality customer experience at modern social platforms. We summarize the main contributions of this work as follows:

- An integrated system to detect fake accounts on Instagram, with utilization of multiple disparate Instagram datasets in a harmonized and enriched feature space for enhanced diversity and generalization.
- A novel feature selection approach, termed Determinantal Submodular Mix for Feature Selection (DSM-FS), which integrates Mutual Information (MI), SHAP-based feature importance, and DPP-inspired diversity preservation to improve feature relevance, interpretability, and selection stability.
- An effective class imbalance treatment using SMOTEENN, applied only to the training set to improve classification robustness, enhance reliability, and prevent data leakage.
- Stacking is an ensemble learning process that combines LightGBM, CatBoost, XGBoost and Logistic Regression for extracting different decision patterns and enhancing prediction reliability.
- Comprehensive analysis of interpretability using SHAP, calibration curves, and threshold scanning to ensure transparency, stability, and reliable dissemination in real world social media environments.
- Experimental validation demonstrating that the proposed DSM-FS + SMOTEENN + Stacking framework achieved 99.71% accuracy and 99.70% F1-score.

It is important to clarify that this work focuses on detecting fake accounts as a binary classification problem and does not include recommendation systems, knowledge graphs, or collaborative filtering methods.

2. Literature Review

The enormous popularity of social media platforms in recent decades has generated a substantial increase in the number of fake and fraudulent accounts that undermine the trust and security of many online communities. As a result, one of the trending research topics in the social network analysis domain has become detecting fake Instagram accounts. This literature review focused on the works of contemporary researchers who greatly motivated our efforts:

Ü Tunç, et al. (2022) they aimed to classify Instagram user profiles into fake, bot, and real accounts with classification algorithms. At the same time, they presented a publicly available dataset for the fake, bot, and real accounts detection on Instagram. Existing self-presented features firstly include conditional features gathered from the three kinds of accounts. Concerning data collection, real accounts are identified and acquired from their circle of friends, fake accounts are obtained by manual scanning as well as Instagram itself, and bot accounts are acquired from purchasing bot accounts websites and mobile applications. The features of these accounts are retrieved by web scraping. They used the seven classifiers to train the classification models of fake, bot, and real profiles detection are as follow. The classifier results explanation can be seen that Random Forest is the highest accuracy in prediction with 90.2%[5]

K. Kaushik et al., (2022) This research aims to develop and implement a deep neural network model trained to detect fake and spam Instagram accounts. It focuses on designing and training a unique neural network model and proposing an algorithm for detecting accounts associated with automated spam messages or fake Instagram accounts. The precision and accuracy of the proposed method were achieved at 93% and 91%, respectively [4].

A.Sallah et al. (2022) They used different machine learning algorithms, such as Bagging and Boosting, to detect fake accounts on Instagram. Machine learning now allows eight people to learn directly from the data rather than human knowledge, with increased accuracy. To balance the two data categories, they used the SMOTE algorithm, which allows for the same number of individuals per category. They also included methods for interpreting complex machine learning models to understand the reasons for model decisions, such as SHAP and LIME values. They preferred using SHAP values because they provide both a local and global explanation of the model, and the values add up to the model's true estimate, which LIME does not. The results show an overall accuracy of 96% for XGBoost and Random Forest. Below, an online fake detection system was developed to detect malicious accounts on Instagram [6].

J. Ezarfelix et al. (2022) Machine learning techniques for detecting fake Instagram accounts were systematically reviewed. Detection will be improved through several approaches, such as image recognition, natural language processing, and metadata analysis. Random forest was found to have the highest accuracy (95.2%) in the subsequent accuracy results, followed by neural networks (94.72%) and naive Bayes (94.58%). Logistic regression and support vector machine (SVM) achieved the lowest accuracy, at 90.82% and 80.43%, respectively. The study suggests

improving machine learning models, incorporating object detection, and improving efficiency to facilitate better integration with social media platforms [7].

P. Azami and K.J. Passi (2024) The researchers used particle swarm optimization (PSO) and binary grey wolf optimization (BGWO) to create a fake Instagram account generator. Faced with the problem of detecting fake accounts, they proposed BGWO and PSO as a hybrid algorithm for detecting fake Instagram accounts. The dataset used consisted of 65,329 Instagram accounts and four classifiers: artificial neural network (ANN), k nearest neighbor (KNN), and logistic regression (LR). The final accuracy reported in the work was 99.1%, with an AUC of 1.0 [8].

S. Chelas et al. (2024) The researchers proposed a machine learning based approach that relies on fewer input features than similar studies. They emphasized the importance of feature selection and optimization, which enhances the reliability and interpretability of the results. Among the models tested, Random Forest emerged as the best performing classifier, achieving a classification accuracy of 97.71% and an F1 score of 95.74%. These results indicate high predictive power, although the slightly lower accuracy suggests room for improvement in reducing false positives. Overall, the study demonstrates that effective detection of fake accounts can be achieved using simplified features, but more extensive validation on diverse datasets is needed to confirm generalizability [9].

H. Gunawan et al. (2025) Several machine learning models were studied to develop a fake account detection system. Individual models, such as support vector machines, naive Bayes, logistic regression, multilayer perceptron models, and ensemble models based on self-clustering and boosting techniques, were trained and tested. The training and testing processes were performed using ten-fold cross validation to prevent overfitting. Test results indicated that the adaptive and gradient boosting models achieved the best accuracy, achieving an F1 score of over 92% and a precision exceeding 93% [10].

Weng et al. (2025) introduced an Educational Data Mining (EDM) system that used a student performance prediction model based on the Gradient Boosting (GB) algorithm to select features. The authors applied CatBoost algorithm to estimate the importance of features and chose the most influential features by applying a cumulative threshold mechanism and repeated 10-fold cross-validations. The framework was tested with Mathematics and Portuguese educational data sets, and tested with different machine learning models such as MLP, SVM, Random Forest, XGBoost and CatBoost. The experimental results illustrated that the proposed feature selection approach showed better prediction results, with the FS-RF model, the best result was obtained in the Mathematics dataset with MAE = 0.9525 and RMSE = 1.5284. The study, however, mainly focuses on tabular educational data and feature importance from CatBoost only, potentially limiting the generalization and interpretability of the framework proposed [11].

Li (2026) proposed that student engagement prediction can be implemented through a multimodal Transformer-based framework, which achieves high accuracy by accurately extracting advanced features from heterogeneous educational data. The study used BERT, Vision Transformer (ViT), and CNN-LSTM architectures to learn textual, visual, and temporal behavioral features from online learning environments. A multimodal Transformer fusion mechanism was utilized to fuse multimodal representations and learn complex student interactions.A cross-modal Transformer fusion mechanism was adopted to fuse multimodal representations and learn complex student interactions. According to the experimental results, the proposed framework has been able to achieve over 90% prediction accuracy while increasing the F1 score in comparison with traditional machine learning techniques. It was based on expensive Transformer models, however, without the use of explainable feature selection or more light-weight optimization methods, which could hinder real-time implementation in educational systems [12].

In contrast, the proposed DSM-FS approach integrates cross-referenced information, SHAP-based significance, and a Determinantal Point Processes (DPP)-inspired diversity mechanism to ensure the selection of informative, non-recurring, and diverse feature subsets [13].

These results imply that in most cases, clustering and optimization as methods for machine learning techniques demonstrate the best performance quality in the detection of fake social media accounts as the studies reviewed finally indicate. However, most of these methods suffer from limitations due to being based on single datasets with very few engineered features, and some cannot properly consider class imbalance. Moreover, few studies consider the interpretability as an important factor to ensure trust and transparency in automated decisions. These voids symbolize the necessity to combine various datasets, state-of-the-art resampling strategies and interpretable feature selection approaches. Motivated by these findings, this work presents a comprehensive and scalable system that improves the accuracy and interpretability of fake account detection.

Table 1: Summary Table of Literature Review

Ref	Dataset	Method	Result
Ü Tunç, et al. (2022)	For data collection, real accounts were determined from their circle of friends, fake accounts were accessed by manual scanning from Instagram, and bot accounts were accessed by purchasing from bot account websites and mobile applications	they used the seven classifiers to train classification models in fake, bot, and real profile detection.	Their results show that the Random Forest gives thehighest prediction accuracy with 90.2%
K. Kaushik et al., (2022)	to develop and implement a deep neural network model trained to detect fake and spam Instagram accounts.	It focuses on designing and training a unique neural network model and proposing an algorithm for detecting accounts associated with automated spam messages or fake Instagram accounts	The precision and accuracy of the proposed method were achieved at 93% and 91%, respectively.
A.Sallah et al. (2022)	fake accounts on Instagram	used various Machine Learning algorithms, such as Bagging and Boosting,	Results show an overall accuracy of 96% for the XGBoost and Random Forest
J. Ezarfelix et al. (2022)	fake accounts on Instagram	Random Forest, Neural Networks, Naïve Bayes. Logistic Regression, SVM	Random Forest was found with the highest accuracy (95.2%) in the consequent accuracy results, followed by Neural Networks (94.72%) and Naïve Bayes (94.58%). Logistic Regression and SVM had the lowest accuracies at 90.82% and 80.43%, respectively
P. Azami and K.J. Passi (2024)	for an Instagram Fake Account Generator	made a combination of PSO and BGWO	the final accuracy reported in work was 99.1% accuracy
S Chelas et al. (2024)	for an Instagram Fake Account	Random Forest	Random Forest emerged as the best performing classifier, achieving a classification accuracy of 97.71% and an F1 score of 95.74%.
H. Gunawan et al. (2025)	Fake account detection	support vector machines, naïve Bayes, logistic regression, multilayer perceptron, and ensemble models based on bootstrap aggregating techniques and boosting.	achieved the best accuracy and an F1 score of more than 92%, with precision surpassing 93%

Traditional machine learning models have yielded good results in previous studies regarding fake account detection. But most of them are based on small or one dataset, some only put attention on Ruin and personal identification data, and commonly utilize basic resampling methods like SMOTE. They also rarely discuss on-the-fly applicability, multimodal feature integration, or hybrid ensemble strategies. To fill in these gaps, this study combines several Instagram datasets into one consolidated, more varied dataset (bundling of a KHMO); employs recent hybrid resampling (SMOTEENN) to address the problem of class imbalance; and develops a stacking clustering model with explainability (SHAP) and feature selection techniques (MI+DPP). Such a combination allows us to be both more performant and interpretable, and better generalizes well across different account behaviours.

3. Proposed Work

In this paper we propose a systematic, transparent and interpretable machine-learning pipeline to differentiate Instagram accounts into genuine vs fake/spam categories. It combines those advanced components in a framework which is quite different from common methods such as Hybrid feature selection (DSM-FS), SMOTEENN to handle class imbalance, and a stacking-based ensemble learning approach. Workflow starts with the integration of data and feature extraction and then proceeds to DSM-FS, which selects informative as well as non-redundant features. The processed dataset is then divided into training and testing sets, where SMOTEENN is only applied to the training data to prevent information leak, facilitating class imbalance help.

Then, you train a set of different base classifiers (LightGBM, CatBoost, XGBoost and Logistic Regression) which are combined in a stacking ensemble model to increase prediction reliability. Finally, the model outputs are evaluated

using multiple performance metrics and further interpreted using SHAP-based explainability, calibration curves, and threshold analysis. The overall workflow of the proposed framework is illustrated in Figure 1.

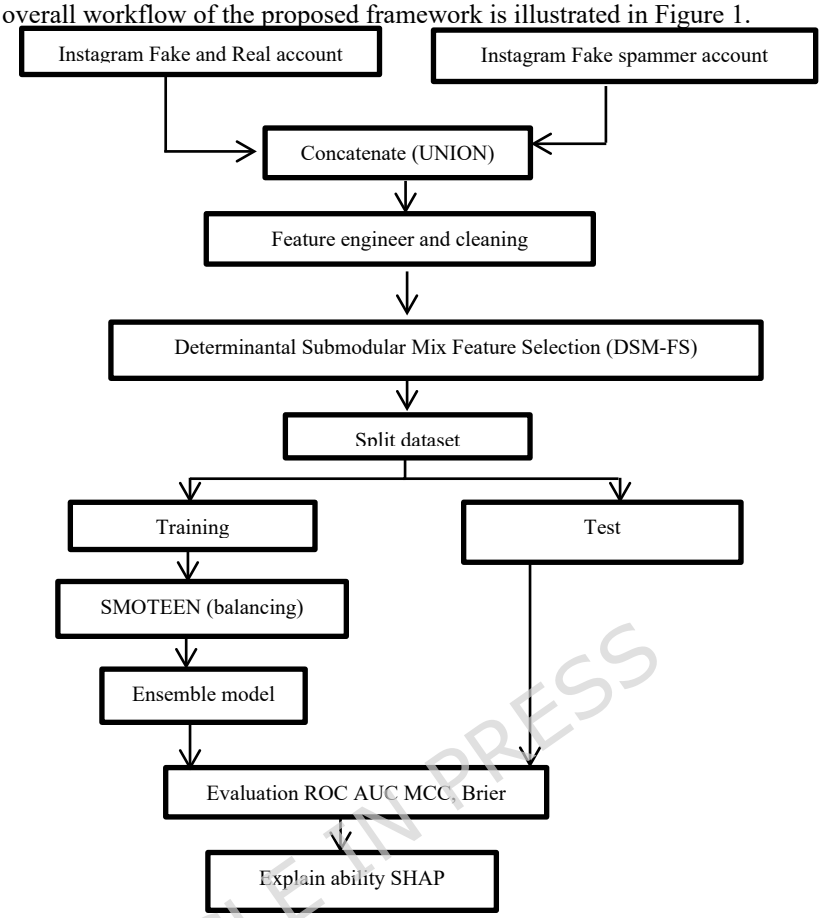

Fig.1. Block diagram of proposed method.

3.1. Description Dataset

In this work, we used the "Instagram Fake and Real Accounts" dataset [14], This record consists of approximately 786 sample Instagram profiles. Each record corresponds to a single user profile and includes several features that describe account behavior and characteristics. First, the features are directly related to the account vanity metrics; the number of followers, edge_count_followed_by, which characterizes user audience size, the number "following" number of accounts being followed by the user at the time of data collection; "edge_count_follow", representing following behavior, and the length of username "username_length." Several binary indicators are also provided, such as whether the username "username_has_number" or full name "full_name_has_number" contains numeric characters, which often indicates auto generated or suspicious accounts. Other features of the vanity metrics are whether the account has a full name length "full_name_length", account privacy toggling on or off "is_private", and authenticity of account creation "is_joined_recently." The dataset also includes whether the account has an Instagram channel "has_channel", is a business account "is_business_account", has Instagram guides "has_guides" or an external URL "has_external_url", often used for promotional or fraudulent purposes. Finally, the target variable "is_fake" identifies whether the account is fake.

The second dataset, titled "Instagram Fake Spammer Genuine Accounts" [15], It includes about 577 sample Instagram profiles. Every object represents a specific Instagram account. The sample is described in terms of behavioral and profile-based characteristics. Features include whether the account has a profile picture; the ratio of numeric characters in a username relative to its length, which is an easily computable feature to identify automatically generated usernames. The data on the number of words in the full name and the ratio of numeric characters in the full name, is relevant due to being relatively natural random; it has already been shown by research. Also, an important feature is a match between the full name and the username: Name = Username. This is a common feature of fake

accounts masquerading as real ones. #Description Length attribute describes the length of the bio / profile description; the "External URL" is a video if a link is present; "Private" is a video if the account is private, or the posts are hidden from everyone except followers; and labels metrics such as the total number of posts. followers, followers.

3.2. Datasets Union concatenate

To expand the feature space and enhance the generalization of the model, two separate datasets extracted from Instagram accounts were combined into a single unified dataset. Namely, both these datasets were first imported, and their graphs were studied. Then the step of unified graph [16] was carried out to achieve uniform naming of columns and data type in both sources, where the features themselves were brought into a single form, while they might correspond to the name of a column username length, number of followers, and privacy state. After harmonizing the datasets, the union of data was performed row wise, preserving all unified features together with the corresponding labels were fake or real. Further, redundant or irrelevant columns, as well as null values, were removed to improve performance. The final result of the union of two datasets made it possible to obtain a more varied and representative sample of Instagram accounts, which became the basis for subsequent feature engineering and training and evaluating the model. This scheme for the Concatenate model of datasets can be presented as:

3.2.1 Schema Harmonization

- Each dataset may have different names for similar attributes (*followers_count* vs. *num_followers*).
- Before concatenation, a harmonization step unifies column names and ensures both datasets describe accounts in the same way.
- The label column is also standardized into a binary form (0 = genuine, 1 = fake).

3.2.2 Column Alignment

- Only the common features (those that appear in both datasets) are retained.
- This ensures the concatenated dataset has a consistent structure: each row represents an account with the same set of features.

3.2.3 Vertical Concatenation

- Once harmonized, the two datasets are stacked row by row.
- Conceptually:
  - Dataset A contributes its records.
  - Dataset B contributes its records.
  - Together they form a larger dataset with the same columns, but more rows (instances).

3.2.4 Unified Target Variable

- After concatenation, the target (label) column is consistent across the whole dataset.
- All training samples now follow the same definition of *fake* vs. *real* accounts.

Once all the joining process is completed, our final dataset has around 1.363 samples, which includes:786 from the first one and 577 from the second counting on. It is a reasonable size of data for traditional ML and ensemble-based methods, though it is not that large.

3.3 Feature Engineering

It should be noted that feature engineering implies a key component of the framework, as the quality and informativeness of the input features have a significant impact on the learning models' final performance [17]. The purpose of this phase is to transform the raw format of Instagram account features to m profound and relevant

representations capable of capturing the behavioral, text based, and network-based trends that characterize fake users from authentic ones.

To achieve that, various categories of features were established. In addition to defining the previously mentioned statistical and relative features across each class, such as the follower-to-follower ratio and Posts to Follower ratio, a large number of additional metrics were computed [18]. For example, there are several accounts that follow several thousand users while being followed only by several ones. Not only were the ratios mentioned above used, but to undo the skewedness of the original data and make the measures of the various features more comparable, other delicate transformations were taken, including z scoring and logarithmic scaling in terms of follower counts [19].

Time and activity feature: This feature accounts for the users' posting time and behavior. The formula used to calculate the total post number of posts per day by the users is given by (Total posts)/(account age in days), where the high value or an irregular post frequency is a sign of spam or automated behavior [20]. Textual and profile feature, this set of features expose the user profile's linguistic and structural features profile about and profile description. When looking into users' biographies and the length of the post, the existence of numbers, tweets, hash tags, emojis, and embedded URLs can give a hint on a spammer. Spammers use links to advertise the products and services and repost the keyword [21].

Similarity features, which use string similarity metrics like the Levenshtein distance [22] and Jaccard similarity [23]. Since imposter accounts imitate existing profiles with slight textual alterations, the above similarity scores indicate imposter profiles are high when over a threshold. Finally, multiple logical tags appended represent the categorical account characteristics like whether the account has a profile picture, privacy status settings, verification badge, business account, etc. These logical tags are like binary characteristics that provide additional context represented by the purely numeric characteristics. This diverse set of geometric features forms a multidimensional feature set for every account: quantitative features reflect the scope levels and activity, while qualitative and logical features indicate the intent and authenticity. Similarly, the similarity features demonstrate the patterns of identity imitation. The DSM-FS administers a standardization process on this feature space created, selecting the optimal subset of the most informative and non-redundant features for the modeling phase.

Finally, several logical tags were added to represent categorical account characteristics, such as the presence of a profile picture, privacy status, verification badge, and business account registration. These binary attributes provide additional contextual information that purely numerical characteristics might overlook.

By combining these diverse types of geometric features, the framework creates a multifaceted representation of each account: quantitative features (such as counts and ratios) capture levels of scope and activity; qualitative and logical features reflect intent and authenticity; and similarity features reveal patterns of identity simulation. This comprehensive feature space is then standardized and passed to the DSM-FS (Determinantal Submodular Mix for Feature Selection) module, which selects the most informative and non-redundant subset of features for the subsequent modeling phase.

3.4 Feature Selection

Feature selection is employed to reduce dimensionality, eliminate redundant attributes, and preserve only the most informative and diverse features for the classification task [24]. In the proposed method, a hybrid approach termed Determinantal Submodular Mix for Feature Selection (DSM-FS) is used. DSM-FS integrates MI [25], SHAP importance values [26], and DPP [27] to balance relevance, interpretability, and diversity. Cross validated F1 score acts as an adaptive stopping criterion. Feature selection can be demonstrated as in Algorithm 1.

3.4.1 Relevance Estimation

For each feature X_j, its statistical dependence on the target y is measured by MI as in (1) [28]:

$$MI(X_j,y) = \sum p(X_j,y) log \frac{p(X_j,y)}{p(X_j)p(y)} \qquad (1)$$

Where X_j Represents the *j*-th feature in the dataset an independent variable whose relevance to the target we aim to measure.

y represent the target variable (label) that the model is trying to predict (such as whether an account is real or fake).

$p(X_j,y)$The joint probability of observing feature X_j and target y simultaneously i.e., how often both occur together in the data.
$p(X_j)$ The marginal probability of observing the feature X_j, regardless of the target value.
$p(y)$ The marginal probability of observing the target value y, regardless of the feature value.

In summary, MI quantifies how much knowing the value of feature X_j reduces uncertainty about the target y. A higher MI value suggests that the feature carries more useful information for predicting the target Simultaneously, a LightGBM classifier is trained, and SHAP values are computed. For feature j, the SHAP importance as in (2) [26]:

$$SHAP_j = \frac{1}{N}\sum \left|\varphi_j^i\right| \qquad (2)$$

Where $SHAP_j$ the average importance (or contribution) of feature j across all samples in the dataset.
N: The total number of samples (instances) considered.
φ_j^i The SHAP value for feature *j* in the *i*-th sample represent how much feature *j* contributed to the prediction for that specific sample.
$\left|\varphi_j^i\right|$ The absolute value of the SHAP contribution, which measures the magnitude of influence (regardless of direction) that feature j had on the prediction for sample iii.
In summary, this formula calculates the mean absolute SHAP value for feature j, which serves as a global measure of how important that feature is across the entire dataset.

Both MI and SHAP scores are normalized to comparable scales. The fusion score is then defined as in (3):

$$q_j = \alpha * MI_j + (1-\alpha) * SHAP_j \qquad (3)$$

q_j The combined importance score for feature j. This score is used to rank or select features based on both relevance and explainability.
α:A weighting coefficient (between 0 and 1) that determines the tradeoff between MI and SHAP importance.
MI_j The MI score for feature j with respect to the target variable. This measures how much knowing feature j reduces uncertainty about the target essentially capturing relevance.
$SHAP_j$ The average absolute SHAP value for feature j, reflecting how much j contributes to the model's predictions in practice capturing interpretability and practical influence.
$(1-\alpha)$ The complementary weight, balancing the influence of the SHAP based importance relative to MI.

In simple terms, the equation produces a hybrid feature importance score by combining theoretical relevance (MI_j) and model-based contribution ($SHAP_j$), weighted by a tunable factor α.

For each feature in the dataset:
Compute a relevance score with respect to the target label
Normalize all relevance scores to a common scale
Return the relevance scores for all features

3.4.2DPP-inspired Diversity Preservation

To avoid redundant features, DSM-FS employs DPP [29]. Let S denote a similarity kernel between features. The L ensemble kernel is as in (4).

$$L = diag(q) * S * diag(q) \qquad (4)$$

Where L: The kernel (or similarity) matrix used in the Determinantal Point Process (DPP) for feature selection. It encodes both the importance of individual features and their pairwise similarities (or correlations).
diag(q): A diagonal matrix whose diagonal entries are the combined feature importance scores q_j (from the earlier formula). Multiplying by this scales the similarities by the importance of each feature.
S: The similarity (or correlation) matrix among the features. It captures how similar or redundant each pair of features is, typically derived from correlation or another similarity metric.

L=diag(q) S diag(q) This constructs a kernel where each entry L_{ij} reflects both the joint importance of features i and j (via q_i) and their similarity (via S).

Classical DPP selects subsets based on their determinants which is computationally impractical for high-dimensional feature spaces due to the exact optimization requirement. Thus, in this work we use a computationally efficient greedy approximation of DPP principles. In particular, marginal gain is employed as a proxy to approximate the maximization of log-determinant objective in order to facilitate both feature relevance and diversity.

This matrix is then used by the DPP to select a diverse yet highly relevant subset of features by maximizing the determinant of the submatrix L corresponding to the chosen features.
For a candidate subset Y, the probability of selection as in (5).

$$P(Y) \propto \det(L_Y) \qquad (5)$$

P(Y): The probability of selecting a particular subset of features Y from the full feature set.
∝: Means "proportional to." The exact probability is a normalized version of the expression on the right.
$\det(L_Y)$: The determinant of the submatrix of L corresponding to the features in subset Y. These determinant measures both the quality (importance) and diversity of the selected features:

- Higher determinant → the subset is both relevant and diverse.
- Lower determinant → the subset is less informative or redundant.

Meaning:

In DPP based feature selection, the probability of selecting a subset Y increases with the determinant of L_Y [30]. This naturally encourages the selection of subsets that balance high feature importance with low redundancy, producing a compact, informative feature set.

Algorithm 1: Diverse Feature Subset Selection using DPP
Input:

- A list of features F
- Relevance scores q_j for each feature
- Correlation matrix C representing feature dependencies

Output: A selected subset of features S⊆FS that balances relevance and diversity
Steps 1: Compute the similarity matrix S by combining correlations and relevance scores:
L=diag(q)·S·diag(q)
Steps 2: Initialize the selected subset as an empty set: ∅=S
Steps 3: While the size of S is less than the desired number of features:

- For each remaining feature f∈F\S:
- Compute the marginal gain in diversity if f is added to S.
- Select the feature with the highest marginal gain.
- Add this feature to S and remove it from the remaining features.

Steps 4: Return the final selected subset S.

3.4.3 Cross Validation Early Stopping

After each candidate feature is added, the selected subset is evaluated using stratified k fold cross validation with F1 score as in (6) [30]:

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \qquad (5)$$

Algorithm 2: Cross Validation Early Stopping
Input: Model, training data (X, Y), max number of iterations, CV folds
Output: The best model parameter
Step1: Initialize best_score = ∞
Step2: For each iteration from 1 to max_iterations:

1. Perform K Fold cross validation on training data
2. Compute average CV score (e.g., F1 or accuracy)
3. If current score > best_score:
    - Update best_score = current score

- Save current model parameters as best_model
- Reset patience counter

4. Else:
   - Increase patience counter
   - If patience counter exceeds threshold:
     - Stop training early

Step 5: Return best_model

Where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$

3.4.4 DSM-FS Algorithm

Algorithm 3: DSM-FS (DPP–SHAP–MINE)

Input: Features X, labels y, base estimator
Output: Selected feature subset S, score table, best_F1
Step1: For each feature j = 1..d: compute MI(X_j; y), train base estimator, compute SHAP importance, normalize scores, and compute fusion q_j.
Step2: Build similarity kernel S (e.g., correlation).
Step3: Form L ensemble kernel L = diag(q) S diag(q).
Step4: Initialize selected set S = ∅, best_F1 = 0.
Step5: While not converged:
a. Choose feature j* with maximum marginal gain in det(L).
b. Update S ← S ∪ j*.
c. Evaluate CV F1 using features in S.
d. If F1 improves: update best_F1 and record S.
e. Else: stop (early stopping).
Step6: Return selected subset S, score table, and best_F1.

3.5 Dataset Split

Finally, the selected features were used to preprocess and engineer the final dataset to be used. The dataset was split into training and testing sets, at a ratio of 80% to 20%. The stratified method was used while splitting to preserve the class distribution. The training set was used to develop the models and run SMOTEENN after the split to oversample the minority classes. The split and resampling method applied ensures that no data from the test set is exposed to the machine learning algorithms. The testing set was used in testing and SMOTE sampled to simulate real world data not seen to this point. This strategy was used to ensure that the models are exposed to real world class distributions, and testing sets are untouched. In addition, stratified k-fold cross-validation was employed within the DSM-FS feature selection stage to evaluate candidate feature subsets using the F1-score criterion. This internal validation strategy improves the robustness of the selected features and reduces the risk of overfitting during the feature selection process. The use of stratified k-fold cross-validation during the DSM-FS feature selection stage provides an internal validation mechanism that improves robustness and reduces the sensitivity of feature selection to a particular data partition.

3.6 Balancing using SMOTEEN

To tackle the problem of class imbalance in the training set, SMOTEENN was implemented in this work. SMOTEENN is a technique that is created by combining SMOTE for Synthetic Minority Oversampling Technique and Enriched Nearest Neighbors. According to the still explained in SMOTE in SMOTEENN: SMOTE constructs new samples to balance the class majority and class minority by randomly selecting a minority candidate and computing the Euclidean distance between the candidate and its k nearest neighbors [31]. This combined approach ensures that the training set is balanced and accurate, improving model performance and reducing bias toward the majority class. The resampled dataset was then used to train all machine learning models individually and within the group.

3.7 Create Ensemble model

Finally, to enhance our predictability, we created an ensemble model that blends several base classifiers. We selected four base learners, which are LightGBM, CatBoost, XGBoost, and logistic regression. All four of these learned on the same SMOTEENN balanced original training data. We then combined their predictions by means of stacking, where the multiple leaners, which was logistic regression, learned on the top of scores performed by the base models. Stacking ensembles exploit the complementary strengths of multiple predictors while mitigating their individual weaknesses; hence, they perform a more accurate and efficient function in our case classification of Instagram accounts as fake or real [32].

3.7.1 Base model

There are several individual classifiers in the core models of our framework, aiming to extract separate patterns from the data. Specifically, as aforementioned, these are LightGBM, CatBoost, XGBoost, and logistic regression. Subsequently, our core learners are trained separately on a balanced training set to predict whether an Instagram account is genuine or fake. However, the individual classifiers have different weaknesses and strengths. The ensemble stacking model thus leverages this diversity of core learners to achieve better overall prediction.

3.7.2 Staking

The stacking process is an ensemble method that links the predictions of a set of multiple base models. In this case, the base learners: LightGBM, CatBoost, XGBoost, and logistic regression, outputs, which are the probabilities of the models' results, are taken as input features to a multiple learner that learns a logistic regression model, which output will be the result. This enables the obtaining of a final prediction for which each base model's strong point is maximized while their respective weak points are mitigated, thus enhancing accuracy and reliability in making a binary decision between fake and real Instagram accounts [33].

For the stacking framework suggested in this paper, all base models (LightGBM, CatBoost, XGBoost & Logistic Regression) go through the same SMOTEENN-balanced training set. In such a scenario, each model learns individually to find the prediction of whether an account is fake or real.

The predictions are gathered into a new dataset by using the predicted probabilities from each base model which serve as input features for the meta-learner in order to build the meta-features. In particular, for each example, a feature vector is created based on the outputs of the base learners. Then meta-features are used to train a Logistic Regression [1] of model that acts as the final decision layer. To avoid overfitting and ensure generalization, the stacking process follows a structured training pipeline where base models are trained first, followed by the meta-learner trained on their outputs. The meta-learner was trained exclusively using outputs generated from the training data, while the independent testing set was reserved only for final performance evaluation. This procedure prevents information leakage and ensures a fair assessment of the stacking framework.

Based on the above description, the stacking process can be formally described as shown in Algorithm 4.

Algorithm 4: Stacking Ensemble Learning

Input: Training data (X, y)
Step 1: Train base models (LightGBM, CatBoost, XGBoost, Logistic Regression) on training data.
Step 2: For each training instance, obtain predicted probabilities from all base models.
Step 3: Construct meta-feature matrix Z where each column corresponds to a base model's output.
Step 4: Train a meta-learner (Logistic Regression) using Z and true labels y.
Step 5: For test data, generate base model predictions and pass them to the meta-learner to obtain final predictions.
Output: Final predicted labels

3.8 Parameter Settings

In this section, we clearly state for reproducibility reasons the main hyperparameters of each model and preprocessing methods. LightGBM for base learners was set with a 0.05 of learning rate, number of leaves = 31 and maximum depth=-1 (no limit) The CatBoost model simply uses 500 iterations, a learning rate of.05 and depth = 6. The hyperparameters of XGBoost classifier were set with a learning rate of 0.05, maximum depth = 6 and using 200 estimators. L2-penalized Logistic Regression with C = 1.0.

For the stacking model, Logistic Regression was used as the meta-learner with default parameters to ensure stability and avoid overfitting.

SMOTEENN was only calculated to the training set to deal with class imbalance. As for the SMOTE component, it used K-nearest neighbors (KNN) with k=5, while edited nearest neighbors (ENN) also used default settings. To ensure that the comparison was fair and consistent, all models were implemented with standard machine learning libraries (sklearn) and trained on the same SMOTEENN-balanced training dataset.

3.9 Evaluation Metrics

The performance of the proposed framework was evaluated using standard classification metrics, including Accuracy [34, 35], Precision [36], Recall [37], and F1-score [38], along with the Confusion Matrix [39]. These metrics were employed to provide an overall assessment of the model's effectiveness in distinguishing between fake and real Instagram accounts. Using multiple evaluation measures ensures a fair and reliable comparison of classification performance, particularly when dealing with imbalanced data.

4 Results and Analysis

4.1 Experiments

Before constructing embeddings from the Instagram datasets, two Instagram accounts datasets were concatenated, and comprehensive feature engineering was performed. The features comprised numerical statistics and textual metrics and correlated to the account metadata. The labels were binary: 0 – real, and 1 – fake/spam. The preprocessed dataset was divided into two sets: training with 80% and exclusion test with 20%. Due to imbalanced classes, SMOTEENN was used in the training set exclusively. Many base learners were fitted: LightGBM, CatBoost, XGBoost, and Logistic Regression. Subsequent fitting and evaluation were performed on both single ensemble and stacking models. Additionally, threshold tuning down the balanced SMOTEENN training set was made to predict better.

4.2 Results

It is important to note that the proposed framework is based on classical machine learning models and ensemble techniques rather than deep learning architectures, which typically require large-scale datasets. Therefore, the dataset size used in this study is appropriate for the employed methods. Furthermore, the integration of feature selection (DSM-FS) and hybrid resampling (SMOTEENN) enhances the effective utilization of available data and improves generalization performance.

Stacked Ensemble: performed best on the exclusion set with high accuracy, F1 score, and balanced precision whereas Single base models performed mixedly; some models had overfitting behavior train on the balanced SMOTE data and SMOTEENN Effect with balancing the training set with a mixed significantly improved model dynamics and a reduction in bias toward the majority class. Feature selection indicated DSM-FS that selected a subset of highly relative and diverse features that improve generalization. Confusion matrices and classification reports graphical summaries generated for all models as ensemble approach consistently reduced misclassification of both real and fake accounts compared to individual learners. This experiment summarized the achievement of combining feature selection, balanced resampling, and ensemble learning in classifying Instagram accounts Confusion matrix of CatBoost classifier distinguishing between real accounts 0 and fake/spam accounts 1. The classification result presented that his model is highly effective at detecting fake account with a very low number of misclassifications of real accounts characterized by low FP. There are many undetected fake accounts (FN=22), and the model is slightly cautious about classifying an account as fake. In conclusion, The CatBoost model achieves highly in accuracy at detecting fake accounts with a very low false alarm rate. Critical balances are required for real world scenarios as minimal reporting of real users is essential such as the social talked about in Figure 2.

The LightGBM classifier has almost the same results in differentiating between real 0 and fake/spam 1 accounts as CatBoost. Both classes' number of accurately and inaccurately classified instances is identical. LightGBM is very accurate and reports only one real account as fake, whereas it misses only 22 fake accounts. This means that the LightGBM model has the same peculiarities as CatBoost – it does not have many false positives, and it is slightly

oversensitive. This situation is acceptable for social media security, where the cost of mistakenly blocking an actual user is much higher than the cost of letting some fake accounts go unnoticed, as shown in figure 3.

Figure 4 displays the confusion matrix for logistic regression. As mentioned above, logistic regression achieved great accuracy for class 0 simply because there are no false positives: it did not make a single mistake by classifying an actual account as fake. This is essential to an application because it is paramount that a log. regression never gets trusted by a user returns fake. However, there were 24 spams it could not classify correctly, which is a bit falser negatives than CatBoost performs. Thus, the model is very conservative: it tries to minimize false positives but concedes some recall for the fake class. In Figure 5, the confusion matrix for the XGBoost model is shown, where the model has a high accuracy distinguishing an actual and a fake post, only having one false positive: it almost never classifies a real post as a fake. Nevertheless, there are 23 false negatives: some fakes pass as reals. Apparently, the model is very good at recognizing an actual post, but further tuning is needed to achieve a smaller number of fake recognition misses. The confusion matrix verifies the good performance of the ensemble staking model, where the number of true negatives 99 is high, and the model does well in identifying the actual posts. There is no one false positive, it almost never has to classify actual posts as fake. In another hand there are only one false negative, and this is an excellent indicator. As for the number of true positives 197, it is high, meaning the model does well in classifying fakes. This confusion matrix was for the testing set after applying SMOTEEN balancing and evaluated at the best threshold (t). The figure shows that the model almost perfectly distinguished between real accounts and fake accounts, as shown in Figure 6.

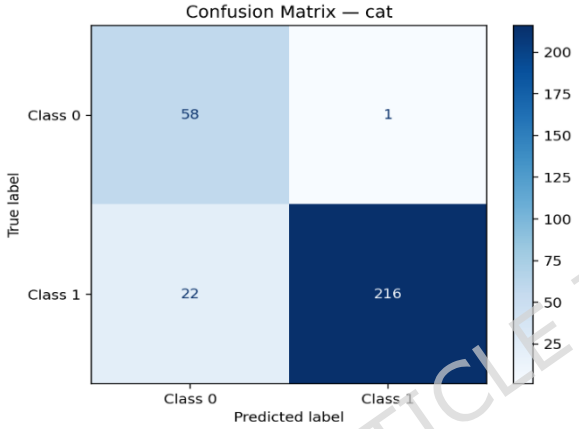

Fig.2. Confusion matrix of cat (with smooteen)

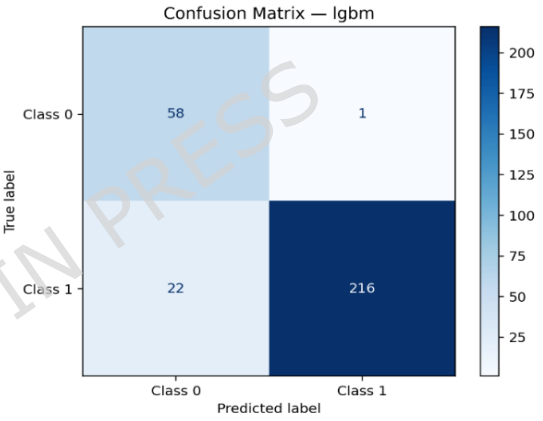

Fig.3. Confusion matrix of LGBM (with smooteen)

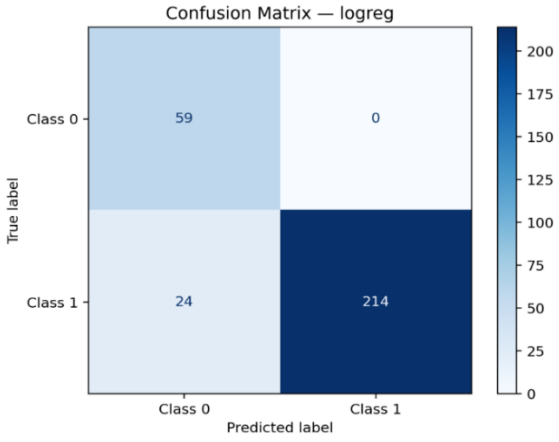

Fig.4. Confusion matrix of logreg (with smooteen)

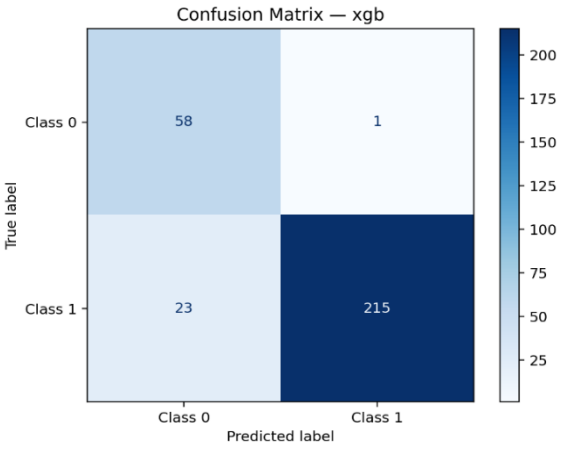

Fig.5. Confusion matrix of XgBoost (with smooteen)

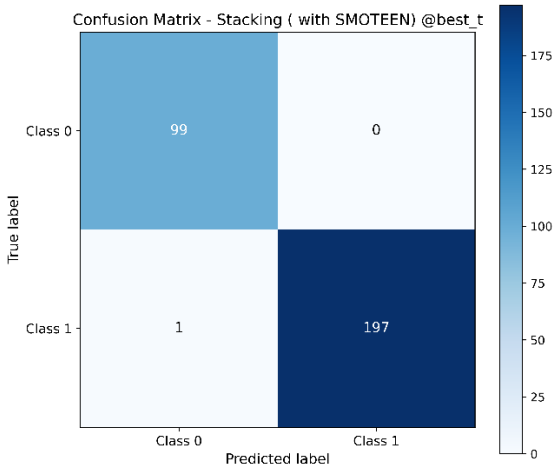

Fig.6. Confusion matrix with SMOTEEN at the best threshold (t).

A calibration curve also created for our proposed last model on the holdout (test) set as shown in Figure 7, where this curve explains the following parameters:

- X axis: Predicted probability that represent the probabilities output by the ensemble stacking model for the positive class 1.
- Y axis: Observed frequency that indicate the actual fraction of positives in the data for those predicted probabilities.
- Orange dashed line (diagonal) denotes to the perfect calibration → predicted probability = actual probability.
- Blue line (Ensemble model): represent how well your predicted probabilities match reality.

The Key insights from the curve:

1. At low predicted probabilities (~0.0–0.1): The observed frequency is slightly higher than the predicted one this means the model underestimates the likelihood of positives when it predicts low probabilities.
2. At mid-range probabilities (~0.3 – 0.4): The curve is above the diagonal, suggesting the model is still under confident, example: when the model predicts ~30%, the actual chance is closer to 75%.
3. At high probabilities (~1.0): The model is very accurate, predictions of ~100% really correspond to almost always being positive.

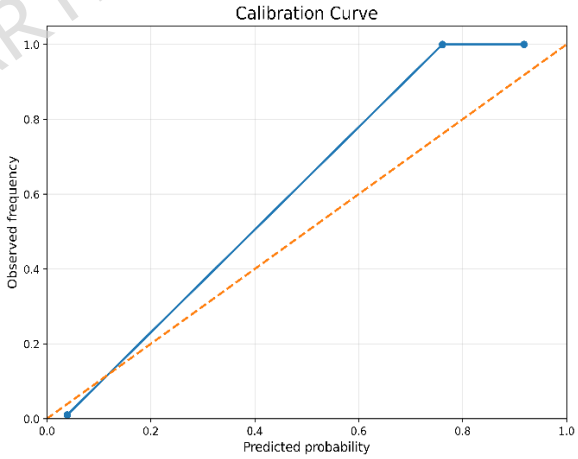

Fig 7. Calibration curve

Threshold scanning of the SMOTEEN balanced test set We evaluate accuracy, F1 score, balanced accuracy at various decision thresholds on the testing data. The experiment results indicate excellent robustness within a wide range of thresholds, where all evaluation metrics are maintained near maximum value, generally over or about 0.99. Only marginal increases in F1 score and balanced accuracy are observed with decreasing threshold values since this

corresponds to higher sensitivity towards the positive class. As the threshold step goes beyond mid-range, we see performance degrading mildly with a sharp drop in accuracy and F1 score due to more false negatives. In general, the model exhibited high reliability against threshold selection; this was reflected in the superior predictive performance observed along most of the threshold values presented in Figure 8.

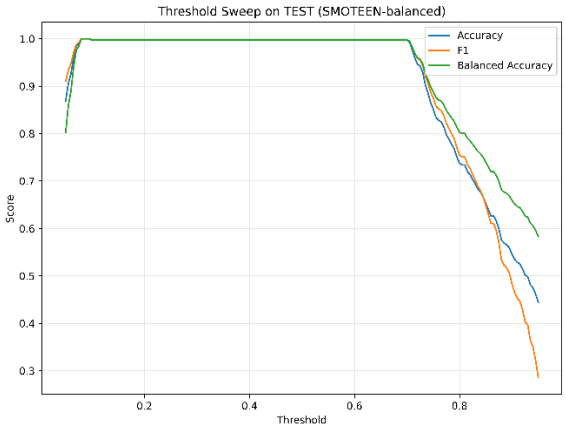

Fig 8. Threshold sweep on testing set with SMOTEEN and ensemble model

In this paper, we presented a SHAP strength graph, which explains the contribution of individual features to a single prediction made by the model. The baseline value (around 0) represents the average output of the model before taking into account any features. The last predicted value is f(x) = 13.87, implying that the model strongly pushed the prediction. The last predicted value f x = 13.87, meaning that the model strongly pushed one class's prediction. In this case, it is "true," as explained in the binary classification. Meanwhile, the red features, located on the left side, push the prediction higher. For example, edge_followed_by and fake had a positive effect while the blue ones, on the right side, push the prediction lower. For example, the is_fake, posts_log 1p, and followers_log1p have significantly gained a lower score. In other words, the account is more likely to be considered as real, as shown in Figure 9.

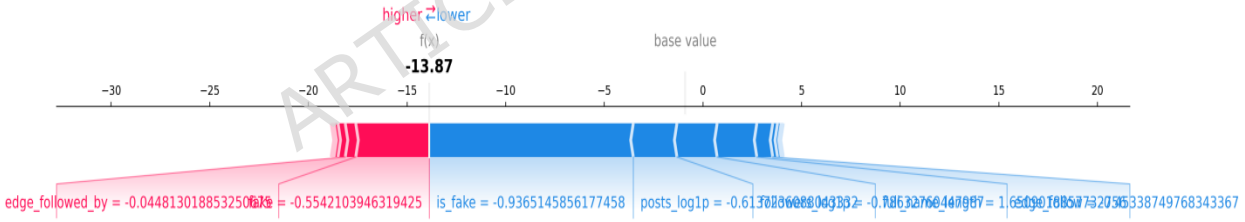

Fig 9. SHAP force plot

The second SHAP plot illustrates the crucial features of the model for detecting fake accounts. As shown in Figure 10, the feature importance analysis reveals that the model's most significant variable is is_fake, followed by engagement. More specifically, a high ff_ratio value and is_fake score strongly signals a fake account and markedly increase its predicted probability, which appears logical since an engagement pattern is an excellent identifier of a fake profile. In contrast, the followers_log1p and edge_follow features substantially increase the predicted probability of an authentic account as they signal a familiar well-balanced engagement. Finally, the username_digit_ratio and full_name_has_number appear to be largely unimportant, and numerical factors have little effect on the classification's overall outcome. Ultimately, the fake classifier depends almost entirely on multiple engagement patterns involving low followers and high followers, thus matching the user's intuition regarding the average behavior of fake or bot profiles.

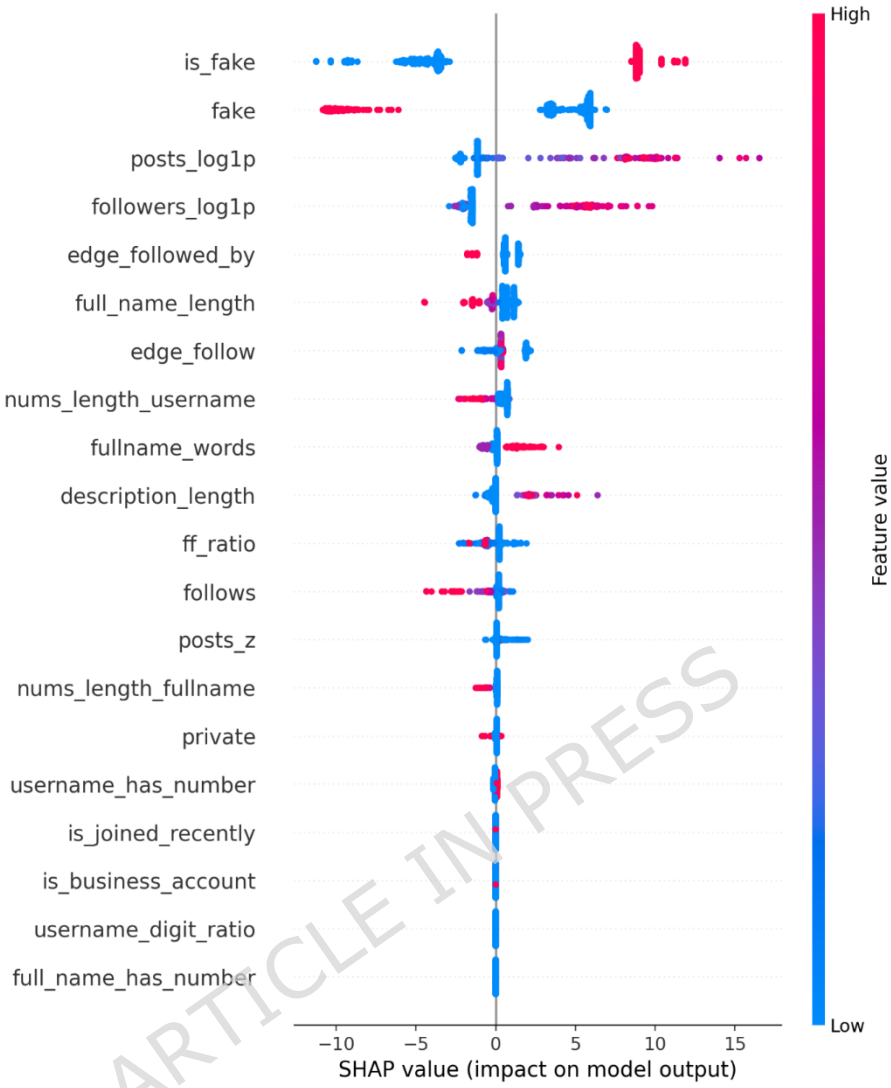

Fig 10. SHAP value

The graph in Figure 11 compares the training and testing accuracies of four machine learning models (Logistic Regression, Random Forest, XGBoost, and LightGBM) without using the stacking technique. Performance: The tree-based models (RandomForest, XGBoost, and LightGBM) significantly outperform Logistic Regression (LogReg) in both training and testing accuracy and in overfitting. Indicator: The large gap between the training accuracy (approximately 100%) and testing accuracy of the tree-based models indicates that they overfit the training data.

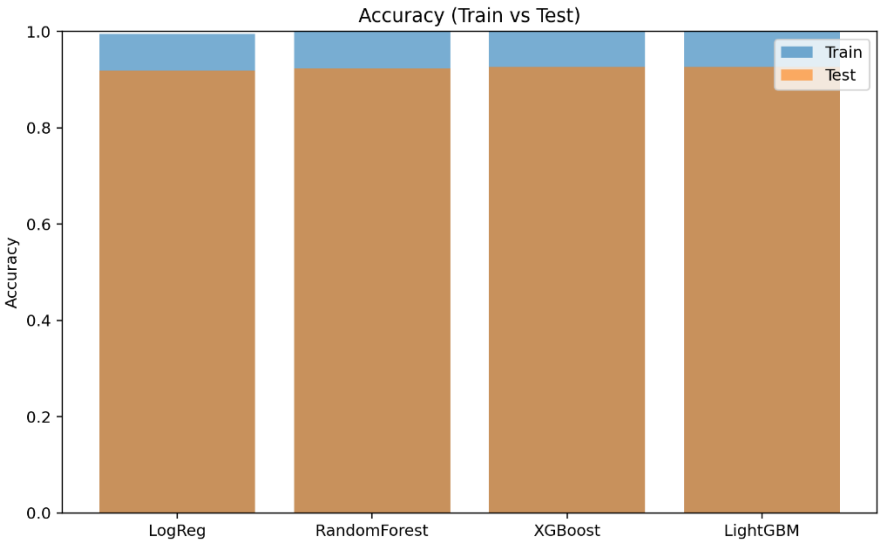

Fig.11. Compares the training and test accuracy of four machine learning models without stacking

The graph in Figure 12 compares the training and test recall scores for the same four models without stacking. Recall measures a model's ability to correctly identify all positive instances (correctly detect fake accounts). A significant drop in recall from the training set to the test set for complex models (RandomForest, XGBoost, and LightGBM) is a clear sign of overfitting. These models memorize the training data but fail to generalize well to new, unseen data. Logistic regression (LogReg) likely shows the smallest gap between training and test recall, indicating that it is the most generalizable model, even if its overall recall score is lower than the other models. This suggests a balance between performance and overfitting.

Though many of the individual models do appear to overfit to the data, as seen by a gap between training and testing performance, the stacking ensemble blends this in part through multiple base learners with different decision patterns. Different models explain various aspects of the distribution of the data, thus combining them mitigates the variance due to model parameters. By the same token, the meta-learner is trained based on results found by base models, the outputs are probabilistic suggestions rather than raw features. This transformation mitigates the overfitting tendencies and enhances generalization performance.

All reported results are based on a fully held-out test set, which was not used at any stage during training or model selection (this makes the evaluation fair).

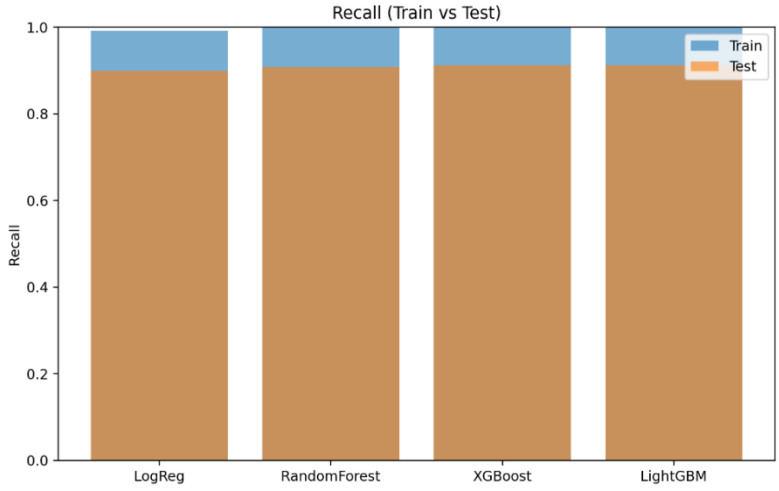

Fig 12. compares the training and test recall scores of the same four models without stacking

The graph in Figure 13 compares the F1 scores (harmonic mean of precision and recall) for the same four models on the training and test data. The tree models (RandomForest, XGBoost, and LightGBM) show high F1 scores on the training data, but much lower ones on the test data. This large gap between the training and test F1 scores is a strong indication that these complex models overfit the training data and fail to generalize effectively. Logistic regression (LogReg) shows the most stable performance with the smallest gap between the training and test scores, confirming that it is the most generalizable model, despite potentially having a lower maximum F1 score.

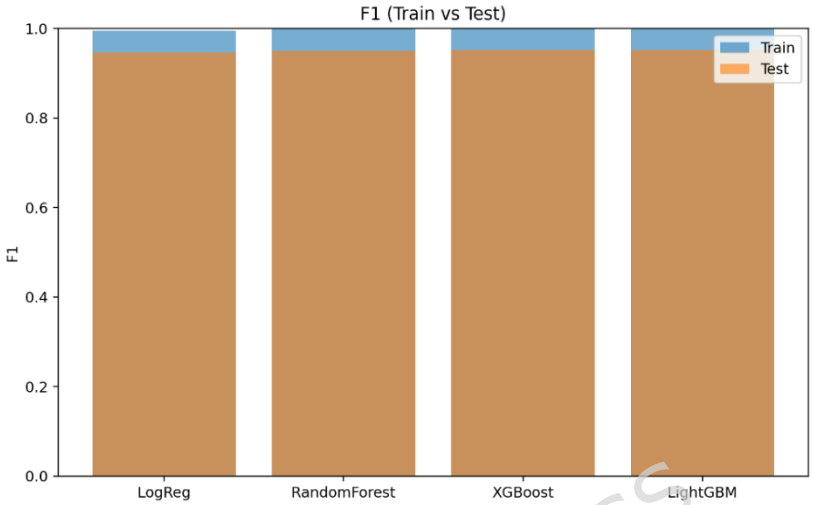

Fig 13. Compares the F1 scores (the harmonic means of precision and recall) for the same four models on training and test data

The bar chart in Figure 14 shows the accuracy scores for both the training and test sets across four models: logistic regression (LogReg), random forest, XGBoost, and LightGBM. The training (blue) and test (orange) bars overlap almost perfectly, with an accuracy of 1.0 for both. This means that all four models achieved perfect accuracy on both the training and test data, never mispredicting a fake account as real (no false positives).

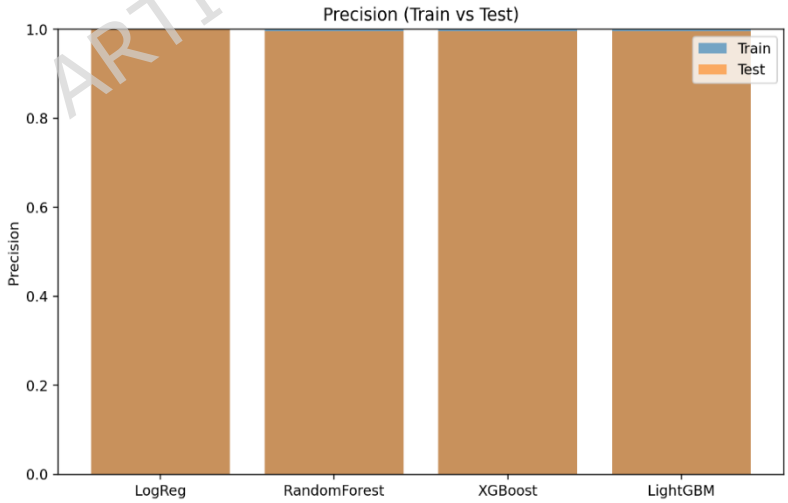

Fig 14. Precision scores for both the training and testing sets across four models

To evaluate the impact of data balancing on model performance, experiments were conducted on both the original imbalanced dataset and the balanced dataset obtained using SMOTEENN. We summarize the pre and post balancing results, demonstrating how resampling techniques improve classification accuracy, F1 score, and overall robustness.

This comparison highlights the importance of addressing class imbalance in fake account detection tasks, as illustrated in Table 2.

Table 2: Results of the ablation study.

Type of Training	Augmentation Method	Algorithms	Results
No stacking	No SMOTEENN and No DSM-FS	Lgbm	Accuracy=91%, F1=94%
		cat	Accuracy=92%, F1=95%
		xgb	Accuracy=93%, F1=95%
		logreg	Accuracy=93%, F1=95%
	With SMOTEENN and No DSM-FS	Lgbm	Accuracy=92%, F1=95%
		cat	Accuracy=93%, F1=95%
		xgb	Accuracy=93%, F1=95%
		logreg	Accuracy=93%, F1=96%
	With DSM-FS	Lgbm	Accuracy=95.8%, F1=96.9%
		cat	Accuracy=96.4%, F1=97.3%
		xgb	Accuracy=96.2%, F1=97.1%
		logreg	Accuracy=95.4%, F1=96.5%
	With SMOTEEN +DSM-FS	Lgbm	Accuracy=97.2%, F1=97.8%
		cat	Accuracy=98.0%, F1=98.3%
		xgb	Accuracy=97.9%, F1=98.2%
		logreg	Accuracy=98.3%, F1=98.5%
Stacking	**With SMOTEENN**	**ensemble model**	**Accuracy=99.71% ± 0.20%, F1=99.70% ± 0.30%**

To assess the stability and reliability of the proposed model, the experiments were repeated multiple times with different random seeds, and the average performance along with standard deviation is reported. The results demonstrate that the proposed stacking framework maintains consistent performance with low variability, indicating strong robustness.

In addition, to put in context the contribution of this paper, the achieved detection accuracy is also compared with that reported in most previous works. The used learning algorithms and optimization methods varied considerably with most of the works reporting high performance. However, the constraints they often faced were the limitation of the dataset, being imbalanced datasets, or lack of interpretability. As a result, the applied method, which combines the aforementioned four methods dataset enrichment, hybrid resampling, selection of features, and stack clustering – is more robust and has superior generalization. Finally, as it can be seen in Table 3 below, the accuracy differences have been demonstrated and the proposed approach proven more effective than the existing methods listed below:

Table 3: Comparison of Classification Results with Recent Studies.

State of Art	Result
Ü Tunç, et al. (2022)	Their results show that the Random Forest gives the highest prediction accuracy with 90.2%.
K. Kaushik et al., (2022)	The precision and accuracy of the proposed method were achieved at 93% and 91%, respectively.
A.Sallah et al. (2022)	Results show an overall accuracy of 96% for the XGBoost and Random Forest
J. Ezarfelix et al. (2022)	Random Forest was found with the highest accuracy (95.2%) in the consequent accuracy results, followed by Neural Networks (94.72%) and Naïve Bayes (94.58%). Logistic Regression and SVM had the lowest accuracies at 90.82% and 80.43%, respectively
P. Azami and K.J. Passi (2024)	the final accuracy reported in work was 99.1% accuracy

S Chelas et al. (2024)	Random Forest emerged as the best performing classifier, achieving a classification accuracy of 97.71% and an F1 score of 95.74%.
H. Gunawan et al. (2025)	achieved the best accuracy and an F1 score of more than 92%, with precision surpassing 93%
Our method	**Ensemble Staking model with SMOTEEN and DSM-FS with the highest predictive accuracy 99.71% ± 0.20% and balanced F1 scores 99.70% ± 0.30%**

Despite the strong performance achieved, it is acknowledged that the high accuracy values warrant careful interpretation. Future work will include additional validation strategies, such as cross-dataset evaluation and ablation studies, to further confirm the robustness and generalizability of the proposed framework.

Conclusion

As a result, this paper proposed a multi-layer framework for detecting fake Instagram accounts. Notably, building from many different datasets into a unified feature space, feature selection, and hybrid resampling raises diversity and representativeness in the identification process. The use of hybrid resampling approaches, such as SMOTEENN, mitigates the problem of class imbalance, as it enhances model learning and removes bias towards the majority samples. We further evaluated different classification models, including logistic regression, random forest, XGBoost, LightGBM, and ensemble stacking. While all models achieved varying performance, stacking outperformed others in more stable and consistent results, indicating the possibility of a good classification accuracy rate based on combining different classifiers. Lastly, feature selection techniques, like MI, SHAP, and DPP, supported the interpretability process, as they allowed for determining the role of each feature in the process. From the findings, the multi-layer approach yield record accuracy and precision compared to monolithic systems, agreed to both individual and ensemble frameworks. In conclusion, this method is scalable, offering a better generalization method across various datasets. Consequently, combining dataset enrichment, hybrid resampling, and ensemble learning and feature selection offers a new method for detecting fake accounts. This study therefore opens horizons for implementation and research; possible future research would be to integrate multimodal data (textual and visual), and real time streaming and feature optimization for large scale implementation.

Funding

This work is supported under Victoria University's Read and Publish agreement.

Author Contributions

Z.A.M.S., A.A.H.A., and F.A.O.S. contributed to data preparation, experimental implementation, and preliminary analysis. M.N.K., K.A., and D.A. developed the methodology, designed the proposed framework, and prepared the main manuscript text. A.I. and A.K.T. supervised the research process, contributed to methodological refinement, and provided critical revisions to enhance the scientific quality of the manuscript. M.N.K. prepared the figures and tables. K.A. contributed to funding acquisition and financial support of the research. All authors reviewed, edited, and approved the final version of the manuscript.

Data availability

The datasets used in this study are publicly available and can be accessed through the following links:

- Instagram Fake and Real Accounts Dataset: https://www.kaggle.com/datasets/rezaunderfit/instagram-fake-and-real-accounts-dataset.
- Instagram Fake, Spammer, and Genuine Accounts Dataset: https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts.

These datasets were utilized for experimental evaluation in this research.

Code availability

Not applicable (custom code)

Acknowledgment

This research was supported by Al-Qadisiyah University, Iraq, and Victoria University, Australia, whose assistance the authors deeply appreciate.

Conflict of Interest

The authors declare no conflicts of interest.

Declarations:

Ethical approval

This research did not involve any human participants, animals, or sensitive data. Therefore, ethical approval was not required.

Consent to Participate

Not applicable.

Consent to Publish

Not applicable.

References

[1] K. Anklesaria, Z. Desai, V. Kulkarni, and H. Balasubramaniam, “A survey on machine learning algorithms for detecting fake Instagram accounts,” in Proc. 3rd Int. Conf. Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 141–144.
[2] F. C. Akyon and M. E. Kalfaoglu, “Instagram fake and automated account detection,” in Proc. Innovations in Intelligent Systems and Applications Conf. (ASYU), Izmir, Turkey, 2019, pp. 1–7.
[3] M. J. Ekosputra, A. Susanto, F. Haryanto, and D. Suhartono, “Supervised machine learning algorithms to detect Instagram fake accounts,” in Proc. 4th Int. Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2021, pp. 396–400.
[4] K. Kaushik et al., “A novel machine-learning-based framework for detecting fake Instagram profiles,” International Journal of Intelligent Systems, vol. 37, no. 9, pp. 7349–7365, 2022.
[5] Ü. Tunç, E. Atalar, M. S. Gargı, and Z. E. Aydın, “Classification of fake, bot, and real accounts on Instagram using machine learning,” Pattern Analysis and Applications, vol. 27, no. 2, pp. 479–488, 2022.
[6] A. Sallah, S. Agoujil, and A. Nayyar, “Machine learning interpretability to detect fake accounts in Instagram,” International Journal of Information Security and Privacy, vol. 16, no. 1, pp. 1–25, 2022.
[7] J. Ezarfelix et al., “A systematic literature review: Instagram fake account detection based on machine learning,” Mathematics and Computer Science Journal, vol. 4, no. 1, pp. 25–31, 2022.
[8] P. Azami and K. J. Passi, “Detecting fake accounts on Instagram using machine learning and hybrid optimization algorithms,” Applied Sciences, vol. 14, no. 10, p. 425, 2024.
[9] S. Chelas, G. Routis, and I. Roussaki, “Detection of fake Instagram accounts via machine learning techniques,” Computation, vol. 13, no. 11, p. 296, 2024.
[10] H. Gunawan, G. S. Budhi, and K. G. Kartawidjaja, “Machine learning-based fake account detection system: Instagram case study,” in Proc. Int. Conf., 2025.
[11] S. Weng, Y. Zheng, C. Zhang, and Z. Liu, “A Gradient Boosting-Based Feature Selection Framework for Predicting Student Performance,” ICCK Transactions on Educational Data Mining, vol. 1, no. 1, pp. 25–35, 2025.
[12] J. Li, “Predictive analysis of student engagement in university physical education courses based on a multimodal transformer algorithm,” Scientific Reports, vol. 16, Art. no. 15123, 2026.
[13] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," vol. 2, no. 1, pp. 56-67, 2020.
[14] Rezaunderfit, “Instagram fake and real accounts dataset,” Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/rezaunderfit/instagram-fake-and-real-accounts-dataset.
[15] free4ever1, “Instagram fake spammer genuine accounts dataset,” Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts

[16] G. Kumar, S. Basri, A. A. Imam, and A. O. Balogun, “Data harmonization for heterogeneous datasets in big data: A conceptual model,” in Proc. Computational Methods in Systems and Software, Springer, 2020, pp. 723–734.
[17] Zouhri, H., & Idri, A. (2024, March). A comparative assessment of wrappers and filters for detecting cyber intrusions. In World Conference on Information Systems and Technologies (pp. 118-127). Cham: Springer Nature Switzerland.
[18] D. D. Javed, N. Z. Jhanjhi, N. A. Khan, S. K. Ray, A. Al-Dhaqm, and V. R. Kebande, "Identification of spambots and fake followers on social network via interpretable ai-based machine learning," IEEE Access, 2025.
[19] K. R. Purba et al., “Classification of Instagram fake users using supervised machine learning algorithms,” International Journal of Emerging Technology and Advanced Engineering, vol. 10, no. 3, p. 2763, 2020.
[20] M. Alizadeh et al., “Content-based features predict social media influence operations,” Science Advances, vol. 6, no. 30, p. eabb5824, 2020.
[21] M. Z. Asghar, S. Habib, A. Khan, and F. M. Kundi, “Social media fake account detection using machine learning: A review,” IEEE Access, vol. 9, pp. 143152–143170, 2021.
[22] B. Berger, M. S. Waterman, and Y. W. Yu, “Levenshtein distance, sequence comparison and biological database search,” IEEE Transactions on Information Theory, vol. 67, no. 6, pp. 3287–3294, 2020.
[23] S. Gupta and A. Sharma, “A comprehensive survey on techniques for numerical similarity measurement in machine learning and data analysis,” Expert Systems with Applications, vol. 270, Art. no. 126648, 2025.
[24] Zouhri, H., & Idri, A. (2025). Feature selection and global interpretability of black-box classification for intrusion detection. Neural Computing and Applications, 37(31), 25835-25866.
[25] Zouhri, H., Idri, A., & Ratnani, A. (2024). Evaluating the impact of filter-based feature selection in intrusion detection systems. International Journal of Information Security, 23(2), 759-785.
[26] H. Wang et al., “Feature selection strategies: A comparative analysis of SHAP value and importance-based methods,” Journal of Big Data, vol. 11, no. 1, p. 44, 2024.
[27] Y. Liu, C. Walder, and L. Xie, “Determinantal point process likelihoods for sequential recommendation,” in Proc. ACM SIGIR, 2022, pp. 1653–1663.
[28] H. Zhou et al., “Feature selection based on mutual information with correlation coefficient,” Artificial Intelligence Review, vol. 52, no. 5, pp. 5457–5474, 2022.
[29] Y. Liu, C. Walder, and L. Xie, “Learning k-determinantal point processes for diverse subset selection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7793–7807, 2022.
[30] Y. Liu, C. Walder, and L. Xie, “Learning k-determinantal point processes for personalized ranking,” in Proc. IEEE ICDE, 2024, pp. 1036–1049.
[31] Zouhri, H., & Idri, A. (2024, November). Assessing the effectiveness of synthetic data generation for Multi-Class Cyber-Attacks detection using generative adversarial networks. In 2024 World Conference on Complex Systems (WCCS) (pp. 1-6). IEEE.
[32] Z. H. Zhou, Machine Learning. Singapore: Springer, 2021.
[33] R. Dey and R. Mathur, “Ensemble learning method using stacking with base learner: A comparison,” in Proc. Int. Conf. Data Analytics and Insights, Springer, 2023, pp. 159–169.
[34] Rfys, R. R., Al-Shammary, D., Kadhim, M. N., & Ibaida, A. (2026). Novel ECG Signal Classification based on Minkowski Distance to Enhance Intelligent Arrhythmia Detection Systems. Smart Health, 100645.
[35] Alzamili, S. L., Baawi, S. S., Kadhim, M. N., Al-Shammary, D., & Ibaida, A. (2025). Efficient feature selection based on Ruzicka similarity for EEG diagnosis. International Journal of Information Technology, 1-15.
[36] Hashim Albohayah, Z. H., Abed, S. B., Mahdi, A. J., Kadhim, M. N., & Najim, A. H. (2025). Ch-PSO: A Novel Embedded Method based on PSO and Chebyshev Distance for Enhanced Epileptic Seizure Classification Using EEG Brain Signals. International Journal of Intelligent Engineering & Systems, 18(5).
[37] Abdulkhudhur, S. M., Abboud, S. M., Najim, A. H., Kadhim, M. N., & Ahmed, A. A. (2025). A Hybrid Deep Belief Cascade-Neuro Fuzzy Approach for Real-Time Health Anomaly Detection in 5G-Enabled IoT Medical Networks. International Journal of Intelligent Engineering & Systems, 18(5).
[38] Baawi, S. S., Kadhim, M. N., & Al-Shammary, D. (2025). Efficient clustering approach based on Gower distance for high-dimensional medical datasets. Cluster Computing, 28(12), 756.
[39] Alrammahi, A. A. H., Sari, F. A. O., Muhammad, Z. A., Kadhim, M. N., Al-Shammary, D., & Ibaida, A. (2025). Enhancing spam detection with advanced feature extraction and unsupervised clustering. International Journal of Information Technology, 1-11.