Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Enhanced XGBoost Machine Learning Model to Detect Fake Social Media Accounts

Pala Prathima[1], Dr. V. Madhukar[2*]
[1]*Research Scholar, Dept. of Computer Science, Chaitanya Deemed to be University, Hyderbad, Telengana.*
E-Mail: palaprathima2021@gmail.com
[2]*Associate Professor, Dept. of Computer Science, Chaitanya Deemed to be University, Hyderbad, Telengana.*
Corresponding Author Email: vmadhukar@chaitanya.edu.in

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Introduction: Online Social media has become an essential part of communication, business, and entertainment in the digital grounds. However, as these platforms growing, the large number of fake accounts are abused that undermine user safety and the trustworthiness of the platforms. These fraudulent accounts are often used for malicious purposes, which poses a serious threat to operations. Therefore, the detection of such accounts is critical for maintaining the integrity of social media platforms.

Objectives: The aim of this research is to develop a faster and more effective method for detecting fake accounts on social media. Given the sophistication of cybercriminal techniques, traditional manual verification and rule-based algorithms are inadequate. This study aims to bridge the gap by leveraging an advanced machine learning approach—specifically, the XGBoost algorithm—to improve the accuracy and efficiency of fake account detection.

Methods: The research employs a modified XGBoost algorithm, combining gradient boosting with L1 (Lasso) and L2 (Ridge) Regularization strategies. These regularization methods help optimize the model's generalization capabilities and prevent overfitting. Additionally, the study incorporates a bagging ensemble method, where multiple models are trained on different subsets of data. This further enhances the model's stability and accuracy. The combination of XGBoost with cross-validation, regularization, and bagging contributes to the detection of fake accounts by minimizing false positives and improving overall performance.

Results: The modified XGBoost model demonstrated a high performance, achieving an accuracy of 94%. The precision, recall, and F1-scores for both genuine and fake accounts were all 0.94. The use of regularization and bagging not only helped mitigate overfitting but also ensured that the model could handle real-world datasets effectively, including those with missing values and skewed distributions.

Conclusions: The research successfully developed an effective machine learning-based method for detecting fake accounts on social media platforms. The XGBoost model, with its regularization and ensemble techniques, significantly improves detection accuracy and reduces false positives, making it a promising solution to address the growing problem of fake accounts. This approach offers a robust framework for real-world application in combating cybercrimes and protecting user trust on social media.

Keywords: Fake accounts, XGBoost, Social media, Ensemble learning, Regularization, Machine learning.

INTRODUCTION

In the modern era, these social media platforms are everywhere and work as important modes for communication, business, entertainment etc [1]. Nonetheless, the surge of various other systems has actually fulfilled a myriad of issues also as well as among these is fake account generation and expansion. These accounts distort the information sharing process, can control user behavior, and are a perfect medium for cyber criminals sabotaging social media efficacy [2]. These are serious issues, impacting users all across the globe that use social media every day to not only stay informed but also have safe interactions [3].

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Detecting fake accounts on social media websites is a complicated issue because of the large amount of data and that malicious entities are designing sophisticated ways to mimic real user behavior [4]. Methods of detection which are based on traditional ways, such as manual verification and simple rule-based algorithms, no longer work for the dimension and complexity of the environment [5]. When it comes to the latter, the traditional way of doing things falls down as nothing stands still and only causes changes in social media tactics to evolve at an even faster rate, meaning that more innovative ways can be used by those creating fake accounts.

Many things are involved in addressing this issue. With the amount of data, we generate on a daily basis; no solution will really be effective if it is not efficient and scalable with the prospected scale from inception. In addition, the fake account behaviors have been mature enough that any detection countermeasures need to be sophisticated and immune from evasion techniques [6]. Solutions are hampered even further by the fact that the real-time processing and rapid response required to combat misinformation can both complicate and delay effective responses. These challenges emphasize the need of novel solutions to any approach that would struggle to keep pace with the rapidly-evolving social media atmosphere for accurately detecting deceitful behaviors.

Within this framework, machine learning (ML) models seem to be a viable answer because they learn from data and hence adapt without the need of explicit programming for every kind of scenario. ML models also detect patterns of behavior that are typically associated with fake accounts by analyzing historical data to identify anomalies and potential threats [7][8]. Using a variety of algorithms and learning on the fly, the thing about machine learning is that it can keep up with devious hackers — this approach is very dynamic. This paper investigates the utilization of machine learning algorithms in detecting social media fake accounts and provides a discussion about their utility and possible future directions.

LITERATURE REVIEW

The research paper introduces a machine learning approach to identify phishing websites in real-time analyzing URL and hyperlink information [9]. The authors developed a novel dataset and applied major machine learning classification techniques, using XGBoost algorithm showing detection accuracy 99.17%. but the proposed method focusing on client side URL and hyperlink features for more effective and real-time solution to phishing detection.

In this chapter the authors address the growing of fake profiles on social media platforms like Twitter, Facebook, Instagram, and LinkedIn[10]. These profiles are often created by humans, bots, or a combination of both (cyborgs) to spread rumors, engage in phishing, data breaches, and identity theft. They discuss various machine learning models that differentiate between fake and genuine profiles based on features such as follower count, friend count, and status updates. They employ datasets like MIB for Twitter profiles and utilize machine learning techniques including Neural Networks, Random Forest, XGBoost, and Long Short-Term Memory (LSTM) networks. The study reports an accuracy of 99.46% using XGBoost and 98% using Neural Networks in detecting fake profiles.

Writer summarized the latest breakthroughs in fake account detection technologies [11]. Then briefly reviewed the challenges and limitations of existing models. The survey serves to help upcoming researchers in recognizing the abstracted areas in the literature and resulting out a generalized outline for fake profile detection on social networking websites.

The authors address on the growing problem of fake accounts in Online Social Networks (OSNs), in many diverse fields [12]. Social media, including Twitter, Instagram and Facebook are falling waist-deep into the net of scamming, as their authenticating standards are weak thus paving way for unlicensed trade to thrive in them. Fake profiles pose a number of threats, carrying out attacks such as phishing attempts, spreading disinformation and fueling social divides. Interpret that as cyberbullying, and deceptive commercial practices. The denotification of fraudulent profiles manually is the biggest time taking and also it creates a lot of frustration & trust issues for users. To judge the authenticity of a profile, social media users usually turn towards the profile picture, description and shared posts to make sure that all these satisfy what is inherent. Many recent studies are devoted to using state-of-the-art machine learning algorithms to extract fake account from different features, such as the profile images and contents shared (fake news or reviews), as well classifying whether a social bot or a normal user is behind this specific account. The paper intends to dive deep into these techniques and compile them, for a better understanding of the state-of-the-art methods available now for future work.

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In this article, review examines the critical aspects of security and trust within online social networks [13]. As online social networks continue to increase, they become increasingly susceptible to various security threats, including profile cloning and Sybil attacks, which can compromise user data and platform integrity. The author highlights the effectiveness and limitation for existing security and trust mechanisms. They also recognize challenges and propose potential solutions to enhance security.

In this article, authors performed profile detection for fake and genuine on Online Social Networks (OSN) [14]. Two datasets of fake and real accounts on Facebook and Instagram were chosen for this purpose. Each dataset contains interrelated features that were investigated through several machine learning algorithms like Naive Bayes, Logistic Regression, Support Vector Machines, K-Nearest Neighbor, Boosted Tree, Neural Networks, SVM kernel and Logistic Regression Kernel.

The authors compared the recent achievements of various machine learning methodologies applied to detect and classify bots on five main social media websites: Facebook, Instagram, LinkedIn, Twitter [15]. It provides a brief survey of all supervised, semi-supervised and unsupervised methods from researchers with their datasets. In addition, it provides detailed breakdown of features extract per categories. A similar quick discussion of the problems and trends are also provided along with future research directions and potential areas for incursions.

The research paper authors presented a new method to identify Twitter spammers by detecting the similarity between spam accounts [16]. This improved the accuracy of three partitioned classification algorithms by injecting some desired features. The proposed method clusters over 200,000 accounts extracted from a random sample of more than 2 million tweets using principal component analysis and then a tuned K-means algorithm to find potential groups of spammers who tweet together.

PROPOSED METHOD

The social media platforms are a principally important channel for communication, marketing and information distribution in recent years. But the problem crops up due to an increasing number of fake accounts which brings inauthenticity and erosion of trust on these platforms. Legacy methods like manual verification and rule-based algorithms are showing their limitations, as most fake accounts now behave in a similar way to legitimate users. Thus, there is more need of advanced machine learning based techniques which would automatically detect and stop the growth of fake accounts in real time.

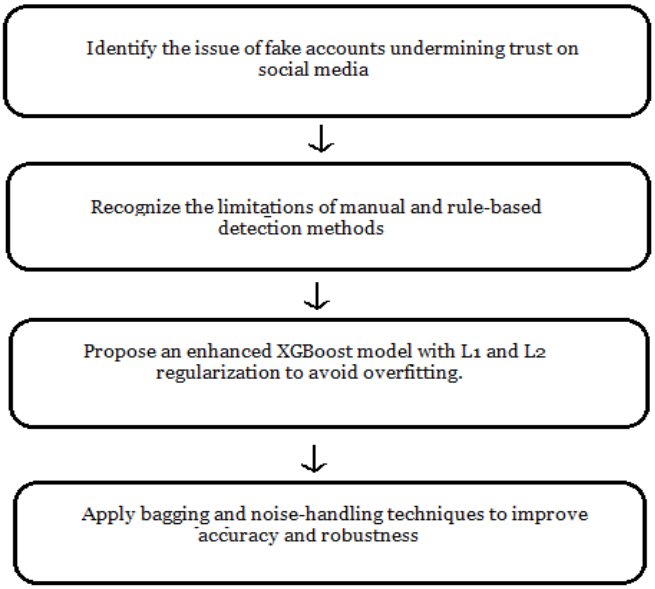

Figure 1: Workflow of Proposed Method

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

To increase the model's accuracy and generalizability, this study proposes a new detection model based on XGBoost machine learning column with regularization techniques and bagging methods. The computation of XGBoost is one of the best part about it because it works very well when confronted to large scale or big data implementations, XGboost with L1 (Lasso) and L2 (Ridge) regularization input prevents overfitting expanding your model complexity. The method reduces variance and leads to more robust models based on bagging, i.e., averaging the predictions from different models trained on different parts of data with an extra effort put in for dealing with noisy and imbalanced data such that the model is able to detect fake account properly. The workflow of proposed method is shown in Figure 1

A. XGBOOST

The XGBoost (Extreme Gradient Boosting) is a powerful and scalable boosting machine learning algorithm. It is ensemble learning method that aggressively builds a sequence of model. It is highly efficient and scalable implementation of gradient boosting machines. However, it is effective when applied to structured or tabular data, making it a go-to algorithm for regression and classification tasks in domains such as finance, healthcare, and marketing.

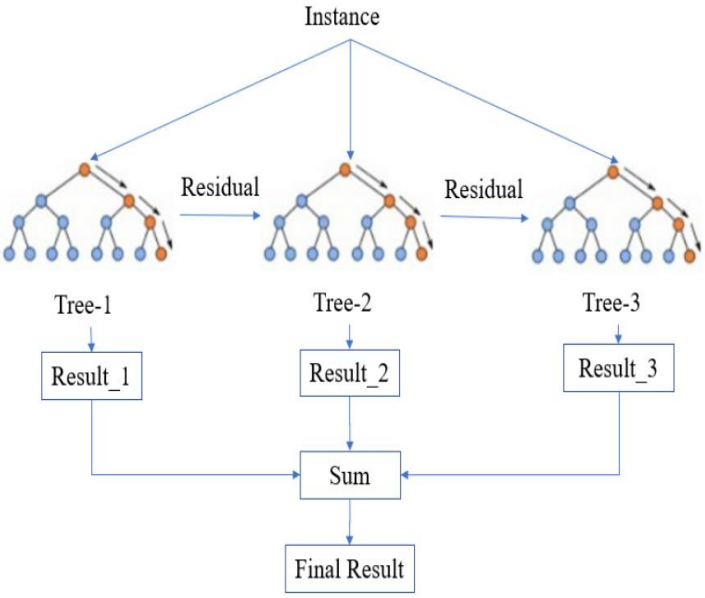

Figure 2. XBoost Architecture

From the Figure 2, we show how XGBoost operates using the boosting ensemble technique, where multiple weak learners—typically decision trees—are sequentially trained to correct the errors of preceding models. This iterative approach optimizes a specified loss function (such as mean squared error for regression or log loss for classification) by following the gradients, hence improving model predictions at each step

B. REGULARIZATION

Regularization is a technique used in machine learning to avoid overfitting which adds a penalty for complex models. Overfitting: If a model has been trained and sudo-performed well on the training data itself, but comes to failing when using new dat. This happens when the model is capturing noise or random fluctuations in your training data instead of the underlying pattern. With the use of a penalty term in objective function, called regularization to combat this issue by its preventive nature. Regularization helps to generalize a model while trying not overfitting it, at the most fitting training data.

There are two most famous regularization types: L1 (Lasso), and L2(Ridge). Adding the absolute values of the coefficients to our loss function is called as L1 regularization, or we can call it also as Lasso. Setting, promoting

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

sparsity in the model, reducing it to itself most important characteristics. However, with L2 regularization — also known as Ridge — the loss function is augmented by squaring the coefficients. Whereas L1 thorughly removes those big coefficients, in contrast to the lest saisfying answer of just decreasing them so much (but not entirely) which is known as ridge regularization. Both techniques are combined to simplify model complexity and improve generalization.

Lasso (Least Absolute Shrinkage and Selection Operator), which is L1 regularization, is an excellent choice for feature reduction These values are penalized by L1 regularization which forces less important features to have a coefficient of zero, effectively removing them from the model. The output is a reduced or sparse model that keeps the most significant features. Lasso is especially useful for datasets that have a large number of variables because many features are unnecessary or repetitive, and with his ability to automatically detect key components and reduce the coefficient by which less important feature.

Ridge regression, or L2 regularization, adds a penalty term to the loss function that is equal to the squared magnitude of the coefficients. In contrast to L1 regularization, L2 regularization penalizes big coefficients more severely in order to prevent any one feature's effect from becoming too dominating. In order to avoid multicollinearity—a situation in which strongly linked characteristics may increase model variance—ridge regression is very helpful. By reducing all the coefficients except for some exceptional ones via L2 regularization, the model becomes more resilient and stable; this further disperses the effect throughout the features.

Elastic Net is a hybrid regularization method combining L1 and L2 forces to regularize weights with both large value frequencies and small value frequency. It has the squared as well as an absolute coefficient values in the penalty term. It is actually a combination of both L1 and L2 regression but generally regarded as a softer form of feature selection given the fact that it allows to select more than just one important variable in the entire process at the same time. It is very useful when dealing with highly correlated features where L1 regularization itself would not do the job. One of the most popular regularized regression techniques is Elastic Net because it provides a good amount of flexibility in terms of regularization and can deal with complex features involving a large number of attributes due to its capability to perform both variable selection and shrinks the coefficient.

Regularization is a key factor in improving generalization ability of machine learning models. Regularization reduces overfitting by preventing the model from being too reliant on the training data set, it does this using large coefficients as a stick. The resulting models are better at generalizing to new data because they focus on getting the main patterns rather than the noise. Regularization blocks the effect of irrelevant features, making model more accurate towards interpretation. While this can be a powerful tool, it is important to adjust regularization well as adding too much of it may lead to underfitting in which the model has not enough complexity and is thus unable to realize the underlying patterns in the data).

C. BAGGING TECHNIQUE FOR ENSEMBLE LEARNING

Bagging or Bootstrap Aggregating is a technique in ensemble learning aims to improve the stability and accuracy of machine learning models. The base idea of bagging is to create multiple copies of a model by training each version on different random subsets of the training data. Each subset is constructed using bootstrapped sampling, which means that some values are picked multiple times in the creation of a subset while others are not selected at all. This is followed by an aggregation of the outputs towards final prediction like voting, averaging, etc. by ensemble. By reducing overfitting and bias, this aggregation makes the model stronger.

One of the main benefits of bagging is that it reduces the variance of the model. A model that fits the training data too closely responds to small changes in the data and has high variance. Bagging reduces the impact of a single training case on the model (which was overfit) by fitting many models to slightly different versions of the data. However, aggregating these individual model-level predictions into a single model makes the entire model more robust and therefore better at generalizing to new data. Bagging is great for helping reduce high variance in models such as decision trees, which tend to be prone hyper variability due to their ability to overfit.

In case of XGBoost, bagging is to create different set of entire model on that subset in the random way and then use these models to predict one record. It will train many XGBoost models out of which combined predictions are used

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

to produce final output. XGBoost is a weak-learner since it performs the boosting technique hence bagging will provide more power to our ensemble model. Bagging can also be used as a noise reduction technique to control the overfitted XGBoost models on noisy datasets ensuring that the final model has lower test error. This approach combines bagging and XGBoost, a powerful ensemble technique that benefits from boosting but also has some extra level of stability thanks to bagging.

Two such important parameters in Bagging when used along with XGBoost are n_estimators and random_state. Parameter n_estimators decides how many base models (XGBoost model) will be in the ensemble. A larger value of n_estimators provides a slightly better performance by improving ensemble's ability to reduce variance but may be computationally costly. This is the "random_state" parameter, which does exactly what you would think: it dictates how random your bootstrapped datasets are. Defining a random_state also allows us to replicate the data in our evaluation of models. With modification to these parameters, one can trade performance for computational efficiency.

While bagging and boosting are both powerful ensemble techniques, the two types of ensembles correct for different kinds of model errors. Whereas boosting is an iterative strategy that reduces bias by acquiring weak learners sequentially, bagging focuses on reducing variance with bootstrapped datasets and training model in parallel. A bagging-boosting hybrid solution has the advantage of variance reduction as wellbias mitigation due to a combination between bagging and boosting (eg. BagXGBoost) The combination makes robust and can be widely used, especially in datasets with high noise or overfitting. Furthermore, bagging's use of ensembles serves as a bit of an equilibrium against outliers or noisy pointsto favor one model over another.

Bagging boosts the generalization of models by averaging out several models trained on different data subsets. For the same model, bagging helps combat overfitting to specific patterns or noise in your data by smoothing out those outliers and creating more balanced predictions as a whole. The generalization also helps a lot in practice since data is not ideal or it will have noise with real-world applications. With XGBoost, bagging gives an edge to its already strong generalization capability making it a little bit more resilient against overfitting.

A potential drawback of bagging is the sacrifice of interpretability. As bagging involves creating an ensemble of models, this makes the decision-making process for the final model difficult compared to a single model. For instance, in XGBoost you can analyze the individual trees of every model for feature importance but when bagging is used its hard to understand which features have more effect on all models. Although this is a type of trade-off that we often have to make in other applications, where predictive accuracy would be more important and knowing how the internal mechanics work can be much less helpful.

Although bagging increases computational costs, it also improves model performance and stability. It takes more time and computing resources to train numerous XGBoost models on various bootstrapped datasets than it does to train a single model. Increasing the value of the n_estimators parameter, which regulates the number of models in the ensemble, results in increased computing demands. Nonetheless, contemporary computer environments—which include parallel and distributed computing frameworks—allow bagging with a high number of models to be executed efficiently. To guarantee that the advantages of bagging offset the higher processing cost, it is important to carefully tune the n_estimators parameter. In reality, this means striking a balance between computational efficiency and model performance.

D. PROPOSED MODEL ARCHITECTURE

The described approach starts with the base model defined as XGBoost which is a powerful gradient boosting approach and utilized with two types of regularization – L1 and L2. This models are designed to prevent overfitting – L1 regularization Lasso shrinks the weights of less important features to zero and thus reduces complexity by significantly reduces the number of features used. L2 regularization, Ridge, punishes large weights and, overall, 'smoothes' the model, making no single feature too important. Using equal strength of these regularizations, it achieves resiliency – it is simple, but complexity is fully controllable and at the same time, its flexibility allows it to adapt to the data.

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Subsequently, the approach integrates a group methodology called bagging, in which many models are trained using distinct data subsets. Reducing variation by averaging the predictions from many models trained on slightly different data samples is the main notion underlying bagging. Ten distinct XGBoost models are trained in this instance, each of which is exposed to a random subset of the training set. Aggregating their predictions together results in a more stable model where little changes in the training set do not lead to huge oscillations and capacity to deal with noisy or imbalanced datasets.

Lastly, when ensemble of model is trained, predictions are made. Due to the ensemble technique used the model has learned from different perspectives and it is unlikely that the overfitting will happen over even single portion of the training data. This can increase the model accuracy and improve robustness and generalization of the mode to new data. This is nice strategy for complex datasets where performance may usually suffer from overfitting and high variance because it encompasses regularization along with bagging

EXPERIMENTAL RESULTS

In following part, detailed results of the simulation obtained by using the proposed technique are discussed. Dataset: The dataset is downloaded from Kaggle, an open-source website.

The dataset is composed of different features that can be used to distinguish between real and fake Instagram accounts like follower count, following count, post engagement, and other behavioural metrics. These features are necessary for framing machine learning models to Instagram spam detecting in general. This projects aims to predict whether an account is fake or a legitimate individual based on them.

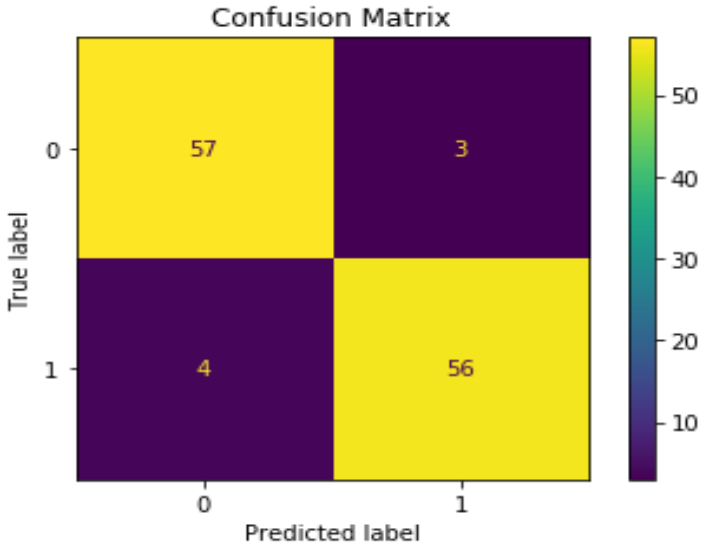

Figure 2: Confusion Matrix

In the confusion matrix shown from above Figure 2, Instagram accounts are classified into two categories: Class 0 accounts is an authentic account and Class 1 accounts are spammer. The following matrix presents the model prediction performance for these two classes with respect of true labels. The four quadrants of the matrix—true positives, true negatives, false positives, and false negatives—represent the various prediction outcomes. These numbers provide information on how well the model differentiates between real and fraudulent accounts.

There are 57 true negatives in the upper-left corner, indicating that the model accurately identified 57 authentic Instagram profiles. Three accounts that were legitimate were mistakenly identified as spammers in the upper-right corner. These are false positives, meaning that the model mistook legitimate accounts for spammers. If a legitimate account is mistakenly reported, it may cause users' annoyance.

The bottom left corner 4 false negatives, here spammers accounts were misclassified as real users. This bug permits a few spammers slip through the cracks, which could cause some damage to network. The accurate identification of

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

spammer accounts is highly important in keeping the platform at a high quality and safety level, so minimizing these false negatives should be one of our first goals to optimize this model.

Finally, the bottom-right corner contains 56 true positives, indicating that the model correctly identified 56 spammer accounts. The true positive value is critical in ensuring that the model efficiently detects spammers. Overall, the model demonstrates a high level of accuracy, as most genuine accounts and spammer accounts were correctly classified, with only a few misclassifications in both directions.

The bottom-right corner, contains 56 true positives which are spammer accounts detected correctly by the model. The absolute number of true positive is really important in practice to make sure the model catches as many spammers. In overall, the model altogether reveals promising accuracy level as most of genuine and spammers are predicted successfully leaving few wrong classifications both ways.

Table 1: Classification Report

	Precision	**Recall**	**F1-Score**
Class 0	0.93	0.95	0.94
Class 1	0.95	0.93	0.94
Accuracy	0.94		

Class 0 = GenuineInstagram Account

Class 1= SpammerInstagram Account

Table 1 shows the classification report that provides an overall picture of how well a machine learning approach can predict if an Instagram account is legitimate (Class 0) or spammer (class 1). The model is evaluated on key metrics that include precision, recall and F1-score have an overall accuracy of 0.94. These metrics give us signal regarding the model efficiency on correct identification of spammers and genuine counts, as well how good are in balancing between false positives (genuine misclassified to spam vs. standouts or true negatives).

Precision measures how many of the accounts predicted to be in a certain class are actually in that class. For Class 0 (genuine accounts), the precision is 0.93, indicating that 93% of the accounts predicted to be genuine are indeed genuine. For Class 1 (spammers), the precision is slightly higher at 0.95, meaning 95% of the accounts predicted as spammers are truly spammers. High precision for both classes indicates that the model effectively reduces false positives, misclassifying only a small number of accounts.

Conversely, recall quantifies how well the model retrieves all true samples of a class. It means that for Class 0, the recall is equal to.95 which in turn represents a percentage of how many genuine accounts are being truly identified it´s equivalent to 95%. Class 1 recall is at 93, meaning the model finds approximately 93% of all spammer accounts. A high recall for both classes indicates that the model is identifying almost all of the real accounts and spammers correctly, however there could be some spammer accounts (true positive) which are missed (i.e., identified as genuine).

F1-score is a single metric that balances the trade-off between precision and recall, which means it takes into account both false positive rate (FP) & false negative rate (FN). Both the classes have a F1-score of 0.94 which shows consistent performance in both class signals. The F1-score close to 1.0 indicates that accuracy and recall are better balanced the model making fewer false positives and negatives in error much as possible This tradeoff is crucial to make sure that the model works nicely on both identifying spammers and not misclassifying authentic accounts.

In the end, the model with an accuracy of 0.94 i.e., it accurately predicted whether messages were real or spammer in 94% cases! As a simple measure, accuracy simply stands for the number of correct predictions divided by total no of prediction made. But in the case of imbalanced dataset accuracy does not provide a whole story, therefore precision and recall are important to assess how well our model is performing for both classes. Clear precision, recall rankings and class levels F1-score display the performance of a model being accurate as well consistent to classify between legitimate Instagram users & spammers.

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Plotting the True Positive Rate (TPR), sometimes referred to as recall, versus the False Positive Rate (FPR) at different classification thresholds, the ROC (Receiver Operating Characteristic) curve in Figure 3 shows how well the classification model performs. The model's ability to differentiate between the two classes—spammers and legitimate Instagram accounts—is shown by the curve. The model performs better the closer the curve is to the upper-left corner. With an AUC (Area Under the Curve) of 0.99, the model performs very well in terms of classification as it is quite good at distinguishing between real and fake accounts. Because it has a high true positive rate and a low false positive rate, a model with an AUC close to 1 is regarded as being extremely accurate

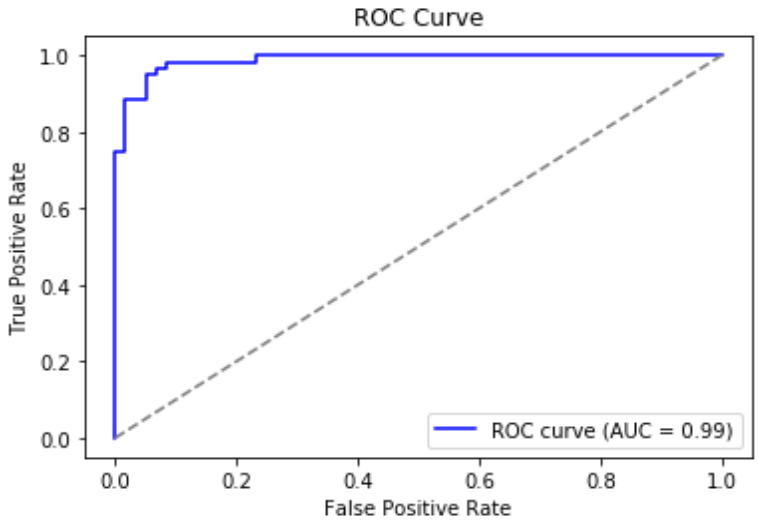

Figure 3: ROC Curve

Figure 4's Accuracy-Recall curve shows how accuracy and recall are traded off for various categorization model thresholds. Recall, or the percentage of real spammers that are accurately recognized, fluctuates, but the curve indicates that the model maintains a high accuracy (almost 1.0) across most recall levels, indicating that a significant number of the accounts predicted as spammers are in fact spammers. Precision somewhat drops as recall rises, indicating the usual trade-off whereby the model may introduce some false positives but catches more genuine positives overall. All things considered, the curve indicates that the model functions well, keeping a good balance between accuracy and recall, particularly in the higher recall ranges. This is crucial for applications that need to limit both false positives and false negatives.

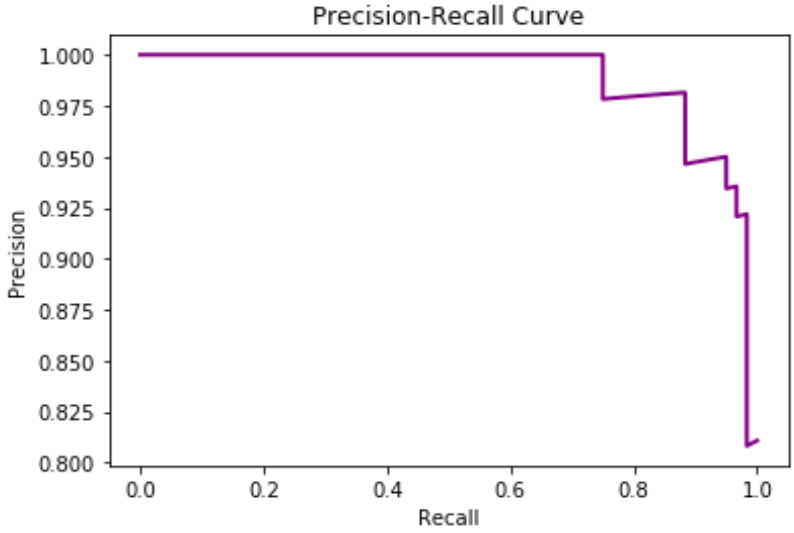

Figure 4: Precision-Recall Curve

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

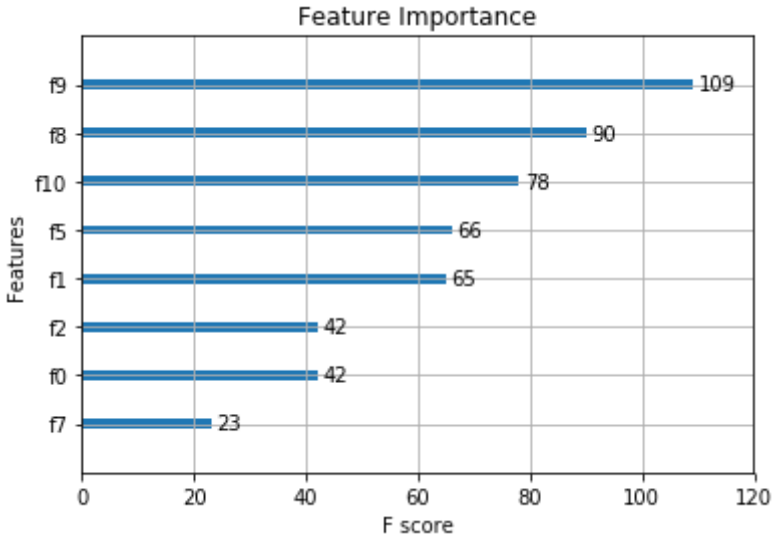

Figure 5: Feature Importance

Figure 5 shows the importance of different features used in building model through bar chart. Feature importance is the measure of how much an independent feature contributes to predicting what you are trying to predict, where larger scores mean more influence. The most important feature, as expected is f9 and the importance score of it is 109 which indicates that how much effect this parameter has on model decision. Other notable features are f8, f10 and f5 having scores of 90,78 and 66 indicating that they most probably have a substantial effect on model predictions as well. The features f0 and f7(fewer important) have lower scores, suggesting that they contributions to the models are small. At the same time, this ranking will help us in interpreting what are features that most correspond to improve our model's accuracy and therefore guide feature selection analysis.

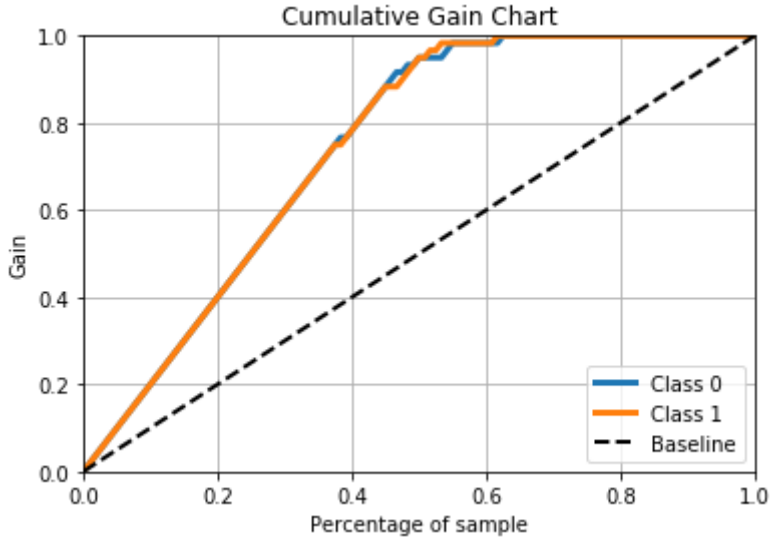

Figure 6: Cumulative Gain Chart

In Figure 6, the Cumulative Gain Chart illustrates how well the classification model performs for both Class 0 (genuine Instagram accounts) and Class 1 (spammer accounts). The chart plots the percentage of the sample (x-axis) against the cumulative gain (y-axis), which shows how much of the positive class (spammers) is captured as the sample size increases. The model performs well for both classes, as indicated by the steep curve rising toward 1.0, meaning that a small percentage of the sample captures a large portion of the relevant accounts (Class 1). The curve for both classes closely follows the ideal diagonal path, significantly outperforming the baseline (random guessing,

Journal of Information Systems Engineering and Management
2025, 10(53s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

represented by the dashed line). This indicates that the model is effective at distinguishing between genuine and spammer accounts, achieving high cumulative gain with a small percentage of the sample.

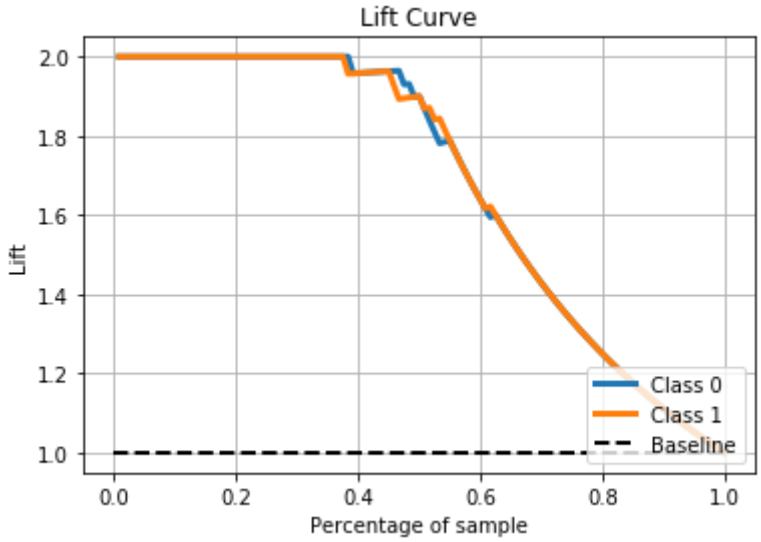

Figure 7: Lift Curve

Figure 7 displays the Lift Curve, which compares the performance of the classification model to random guessing for both Class 0 (real Instagram accounts) and Class 1 (spammer accounts). The lift, shown by the dotted line at 1.0, is the model's performance ratio relative to the baseline, and it is plotted on the y-axis. As you can see from the start, when compared to random guessing, if its lift is 2.0 then the model will be twice as good at picking an accurate class (genuine or spammer). The lift sequentially degrades as the sample % increases, indicating that the model performance is approaching chance with additional information. Yet for most of the sample, the model continues to hold its lift greater than 1.0, which suggests that it beats random selection in both classes often.

CONCLUSION

It is not based It's a new model that includes regularisation and ensemble learning to leverage the weaknesses from the other models. Using this integrated machine learning framework, the model was able to outperform standard detection methods, creating growing and robust fake account-detection techniques. Results show that the model not only enhances detection performance but also decreases both false positives and negatives, which demonstrates its great potentiality in practical scenarios. These results also serve as an important step towards trust and safety enhancement in social media platforms that are plagued by impersonation problems, such as fake accounts. The proposed machine learning model was efficient in detecting fake social media accounts, with an accuracy of 94% proven by this study. Important aspects of the model, such as XGBoost with L1 and L2 regularisation techniques served to prevent overfitting and generalise the model more.

REFRENCES

[1] R. R. Chen, R. M. Davison, and C. X. Ou, “A symbolic interactionism perspective of using social media for personal and business communication,” *Int. J. Inf. Manage.*, vol. 51, p. 102022, Apr. 2020, doi: 10.1016/j.ijinfomgt.2019.10.007.

[2] H. Chen, Y. Yang, and Y. Wu, “Invalid Message Risks and Analysis of Laws to Restrict Cyber Crime in Social Applications,” in *Proceedings of the 2022 11th International Conference on Networks, Communication and Computing*, New York, NY, USA: ACM, Dec. 2022, pp. 341–347. doi: 10.1145/3579895.3579946.

[3] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, “Online social networks security and privacy: comprehensive review and analysis,” *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2157–2177, 2021, doi: 10.1007/s40747-021-00409-7.

[4] B. Guo, Y. Ding, L. Yao, Y. Liang, and Z. Yu, “The Future of False Information Detection on Social Media,” *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–36, Jul. 2021, doi: 10.1145/3393880.

[5] S. Rao, A. K. Verma, and T. Bhatia, “A review on social spam detection: Challenges, open issues, and future

Research Article

directions," *Expert Syst. Appl.*, vol. 186, p. 115742, Dec. 2021, doi: 10.1016/j.eswa.2021.115742.

[6] L. Caviglione *et al.*, "Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection," *IEEE Access*, vol. 9, pp. 5371–5396, 2021, doi: 10.1109/ACCESS.2020.3048319.

[7] M. Rabbani *et al.*, "A Review on Machine Learning Approaches for Network Malicious Behavior Detection in Emerging Technologies," *Entropy*, vol. 23, no. 5, p. 529, Apr. 2021, doi: 10.3390/e23050529.

[8] K. K. Bharti and S. Pandey, "Fake account detection in twitter using logistic regression with particle swarm optimization," *Soft Comput.*, vol. 25, no. 16, pp. 11333–11345, Aug. 2021, doi: 10.1007/s00500-021-05930-y.

[9] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," *Ann. Data Sci.*, vol. 11, no. 1, pp. 217–242, Feb. 2024, doi: 10.1007/s40745-022-00379-8.

[10] U. D. Joshi, Vanshika, A. P. Singh, T. R. Pahuja, S. Naval, and G. Singal, "Fake Social Media Profile Detection," in *Machine Learning Algorithms and Applications*, Wiley, 2021, pp. 193–209. doi: 10.1002/9781119769262.ch11.

[11] P. K. Roy and S. Chahar, "Fake Profile Detection on Social Networking Websites: A Comprehensive Review," *IEEE Trans. Artif. Intell.*, vol. 1, no. 3, pp. 271–285, Dec. 2020, doi: 10.1109/TAI.2021.3064901.

[12] A. K. M. Rubaiyat Reza Habib, E. Elijah Akpan, B. Ghosh, and I. K. Dutta, "Techniques to Detect Fake Profiles on Social Media Using the New Age Algorithms - A Survey," in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, Jan. 2024, pp. 0329–0335. doi: 10.1109/CCWC60891.2024.10427620.

[13] N. Hajli and X. Lin, "Exploring the Security of Information Sharing on Social Networking Sites: The Role of Perceived Control of Information," *J. Bus. Ethics*, vol. 133, no. 1, pp. 111–123, 2016, doi: 10.1007/s10551-014-2346-x.

[14] A. Mughaid *et al.*, "A novel machine learning and face recognition technique for fake accounts detection system on cyber social networks," *Multimed. Tools Appl.*, vol. 82, no. 17, pp. 26353–26378, Jul. 2023, doi: 10.1007/s11042-023-14347-8.

[15] M. Aljabri, R. Zagrouba, A. Shaahid, F. Alnasser, A. Saleh, and D. M. Alomari, "Machine learning-based social media bot detection: a comprehensive literature review," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, p. 20, Jan. 2023, doi: 10.1007/s13278-022-01020-5.

[16] K. S. Adewole, T. Han, W. Wu, H. Song, and A. K. Sangaiah, "Twitter spam account detection based on clustering and classification methods," *J. Supercomput.*, vol. 76, no. 7, pp. 4802–4837, Jul. 2020, doi: 10.1007/s11227-018-2641-x.