Fraud Detection in Financial Transactions: An Ensemble Learning Framework for Risk Management

Jyotiranjan Rout[1], Dr. Uttam Panda[2], Dr. Rasmilata Nayak[3], Saroj Kumar Patra[4]
[1] *Department of Computer Science & Engineering, Balasore College of Engineering and Technology (Affiliated to Biju Patnaik University of Technology, Rourkela, Odisha)*
[2]*Department of Chemistry, Balasore College of Engineering and Technology, Sergarh, Balasore–756060, Odisha, India (Affiliated to Biju Patnaik University of Technology, Rourkela, Odisha)*
[3]*Department of MBA, Balasore College of Engineering and Technology, (Affiliated to Biju Patnaik University of Technology, Rourkela, Odisha)*
[4]*Department of Civil Engineering, Balasore College of Engineering and Technology, (Affiliated to Biju Patnaik University of Technology, Rourkela, Odisha)*

Abstract - The study investigates the effectiveness of several machine learning models in identifying both legitimate and fraudulent transactions. A thorough comparison study reveals that the ensemble model is the best strategy, with exceptional performance metrics across numerous assessment criteria. The ensemble model achieves an accuracy of 0.99, a precision of 0.990, a recall of 0.99, and an F1-score of 0.98, suggesting excellent ability to reliably identify both positive and negative examples while reducing mistakes. The confusion matrix analysis verifies the ensemble model's accuracy and recall, with a low number of false positives and negatives. Furthermore, the ensemble model has an ROC of 0.99, exceeding models such as Random Forest (ROC 0.97), Gradient Boosting Algorithm (ROC 0.96), and Naive Bayes (ROC 0.94), exhibiting higher class distinguishing capacity. While other models perform admirably, the ensemble model's ability to combine the strengths of multiple algorithms leads to significantly improved predictive accuracy and robustness, making it the preferred choice for detecting fraudulent transactions in real-world scenarios.

Keywords: Machine learning, Ensemble model, Fraud detection, Transaction classification.

I. INTRODUCTION

Currently, there is a significant surge in both online and offline commerce. Thanks to substantial developments in IT technologies, including networking, electronic payment, and mobile computing services, electronic markets and online transactions are rising. The phenomenon has increased the prevalence of financial fraud offences, encompassing online and offline activities (Çığşar & Ünal, 2019). Fraudulent activities have substantial adverse economic and societal repercussions on a global scale. Financial fraud has significantly impacted the overall structure of the market economy, leading to a loss of trust and economic harm for customers, investors, and financial institutions. Furthermore, it has led to a widespread decline in trust in the ethical conduct of companies (Albizri et al., 2019). Many investors, policy regulators, and decision-makers are endeavouring to design effective fraud mitigation measures to minimise the repercussions of fraudulent activities and uphold the integrity of financial markets.

The perpetual issue lies in effectively differentiating fake financial data from genuine data. This challenge is of great interest to scholars, norm setters, regulators, audit firms, and investors [3]. Financial fraud detection studies have entered a new stage due to the fast advancement of business intelligence and extensive data analysis methodologies. From a data analysis standpoint, the primary objective of fraud detection is to utilise data mining algorithms to find patterns of fraud or anomalies within extensive financial transaction records. To accomplish this objective, numerous researchers have put forth inventive methodologies, algorithms, and detection techniques. However, as fraud detection tools have advanced, fraudsters have also adapted their fraudulent methods to evade detection (Afriyie et al., 2023). According to Baesens et al. (2021), the use of data mining tools is necessary for detecting fraud. Therefore, it is crucial to emphasise the need for continuous innovation in fraud detection technologies (Faraji, 2022).

Data mining approaches have demonstrated their

utility in various areas within this field, including credit card approval, bankruptcy prediction, and market analysis (Rathore et al., 2021). Logistic regression and neural network algorithms have been widely employed in previous studies on financial fraud detection (West & Bhattacharya, 2016). Sohl et al. (1995) (Sohl & Venkatachalam, 1995) were the first to utilise neural network technology to detect financial statement fraud. In the same year, Beneish & Vorst (2022) used the logistic regression technique for fraud investigation and identification. (Omar et al., 2017) employed a neural network that used publicly available financial data to identify deceptive financial statements. (Hilal et al., 2022) presented several statistical fraud detection techniques that rely on statistical learning.

However, the efficacy of the majority of current approaches remains suboptimal. Despite the widespread use of various techniques, most current methods for detecting financial fraud are still insufficient in achieving satisfactory levels of fraud detection (Abbasi et al., 2012). This underscores the significant challenge that the detection of financial fraud poses for business intelligence solutions and the pressing need for improvement and advancement in this field.

II. LITERATURE REVIEW

Some researchers employed the M-Score (Handoko & Natasya, 2019) (Kamal et al., 2016) (Maniatis, 2022), F-Score (Ratmono et al., 2020), and Z-Score (Ofori, 2016) models early on to assess the likelihood of financial fraud. A few academics have also confirmed the applicability of Benford's rule by applying it to the identification of financial fraud in the accounting field using the dataset's first-digit distribution law (Arisa et al., 2018).

As artificial intelligence technology has advanced so quickly in recent years, many academics have been using machine learning (ML) algorithms like logistic regression (LR) (Yusrianti et al., 2020), back propagation neural networks (BPNNs) (Xiong et al., 2022), support vector machines (SVMs) (Cao & Liu, 2019), and decision trees (DTs) (Eweoya et al., 2019) in the field of financial fraud identification. Research on the identification of financial fraud has also used various deep-learning systems. Hierarchical self-attention (HSA) (Ruan et al., 2021), long short-term memory (LSTM) (Jan, 2021), self-organising maps (SOMs) (du Jardin, 2016), and convolutional neural networks (CNNs) (Hosaka, 2019) are a few examples. Many academics have focused on ensemble-learning algorithms because single-classifier models have limitations imposed by the models, and performance improvement has reached a breaking point.

Today's more sophisticated ensemble learning methods mainly consist of boosting, stacking, and bagging algorithms. Additionally, several academics have suggested hybrid prediction models (Ijaz et al., 2018; Guleria et al., 2022). The bagging algorithm uses a random sampling technique to create many single classifiers, making a final classification result through voting; in other words, the minority submits to the majority. The random forest (RF) algorithm, based on the DT, is a typical example of a bagging algorithm. The RF model was superior to single-classifier models like LR, SVMs, and DTs when they used the RF algorithm to detect financial statement fraud (Ye et al., 2019). Later, utilising an improved RF algorithm, An and Suh developed a financial statement fraud identification model demonstrating superior classification performance (An & Suh, 2020). The boosting and bagging algorithms are comparable; the main distinction is that the boosting strategy uses serial iterative execution for single classifiers, whereas the bagging technique executes single classifiers in parallel. To enhance identification accuracy and decrease training errors more significantly, distinct weights are assigned to each classifier simultaneously. The three primary methods for boosting are gradient boosting decision tree (GBDT) (Du et al., 2020), extreme gradient boosting (XGBoost) (Lin & Bai, 2022), and adaptive boosting (AdaBoost)(Bao et al., 2020). Still, the stacking method combines single classifiers through a new learner, whereas the bagging and boosting algorithms integrate single classifiers by simple voting or weighted voting. Pisula employed a stacking ensemble model to forecast the insolvency risk of Polish enterprises (Pisula, 2020). Additionally, Liang et al. built a model for predicting bankruptcy based on the stacking ensemble model, showing that the stacking ensemble method had a more substantial recognition impact (Liang et al., 2020). It does not demonstrate how the stacking ensemble learning model may be used in accounting research. Still, it does offer experience using the algorithm to identify financial fraud.

III. BACKGROUND

3.1 Random forest

Random Forest is an approach for supervised machine learning based on ensemble learning. An "ensemble learning" method generates predictions by repeatedly assembling or bagging similar models. The name "Random Forest" comes from how the process, which employs many decision trees, operates similarly and produces a forest of trees. The random forest algorithm applies to classification and regression applications (Kumar et al., 2019).

The supervised machine learning algorithm Random Forest uses a collection of decision tree models for classification and prediction (R, 2015). Due to their low predictive capacity, decision trees are all weak learners. Its foundation is ensemble learning, which classifies an issue and increases the model's accuracy using numerous decision tree classifiers (Kabir et al., 2018). The random forest uses the bagging approach to create a forest of decision trees. The random forest algorithm takes a dataset (X, Y) with N total observations, where X is the outcome variable, and $\boldsymbol{V}$ is the predictor variable. It first generates Ki random variables ($i = 1, 2, \ldots, N$) to form a vector, and then it transforms each Ki random vector into a decision tree to obtain the dKi decision tree ($dK1$ (X), $dK2$ (X), …, dKN (X)). The following are the final classification results:

$$D(X) = \arg\max_{i} \left\{ \sum_{N} d\,K_i(X) = Fraud, \sum_{N} d\,K_i(X) = Not\ fraud \right\}$$

A feature selection process is usually unnecessary when using a random forest (Kabir et al., 2018). This approach's shortcoming is how soon it could flag as false data that has an extensive range of values and variables with multiple values. According to Tasnim et al. (2022), it is one of the financial sector's most accurate fraud detection algorithms. The Random Forest approach typically introduces more significant uncertainty when building the tree, so it is critical—especially in node splitting—to select the most significant feature for analysis out of all the features. A random forest algorithm technique is shown in Fig. 5.

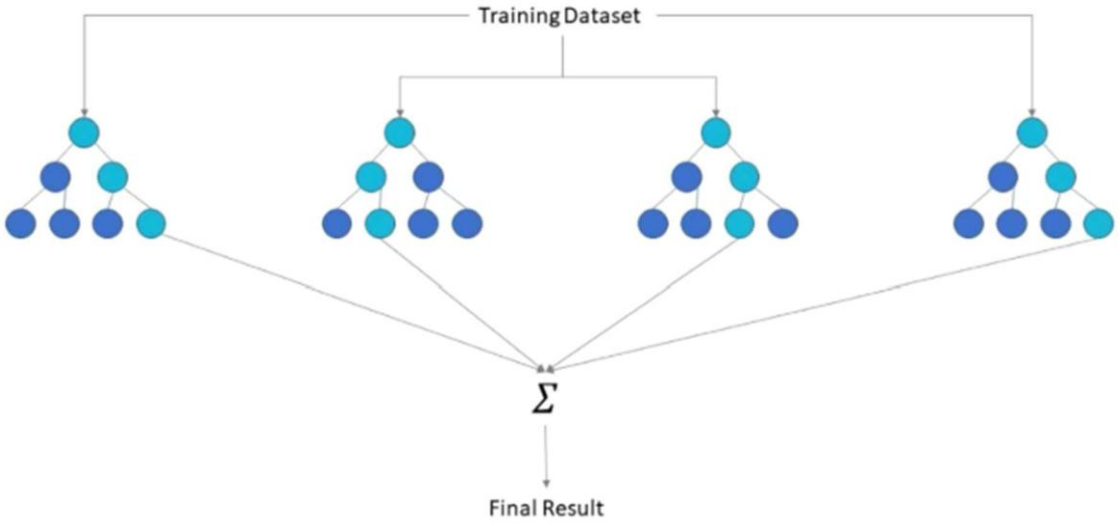


Figure 1: Random forest

3.2 Gradient Boosting Algorithm

Boosting algorithms iteratively combine weak learners, which are learners that are marginally better than random, to create a strong learner. Gradient boosting is an approach that is similar to boosting and is used for regression tasks. The objective of gradient boosting is to obtain an approximation, , of the function that translates instances x to their output values y, using a training dataset. This is achieved by minimising the expected value of a specified loss function. Gradient boosting constructs an incremental estimation of a target variable by combining many functions in a weighted manner.

where p_m is the weight of the m^{th} function, $h_m(X)$. These functions are the models of the ensemble (e.g. decision trees). The approximation is constructed iteratively. First, a constant approximation of $F^*(X)$ is obtained as

$$F_0(X) = \arg\min_{\alpha} \sum_{i=1}^{N} L(y_{i,\alpha}).$$

Future models are anticipated to reduce

$$(p_m, h_m(X)) = \arg\min_{p,h} \sum_{i=1}^{N} L(y_i, F_{m-1}(X_i) + p h(X_i)).$$

However, rather than directly addressing the optimisation issue, each h_m step may be seen as a greedy iteration in a gradient descent optimisation for F^*. Each model is trained on a fresh dataset $D = \{X_i, y_{mi}\}_{i=1}^{N}$, where the pseudo residuals y_{mi} are computed.

$$y_{mi} = \left[\frac{\partial L(y_i, F(X))}{\partial F(X)}\right]_{F(X)=F_{m-1}(X)}$$

The value of p_m is determined by solving a line search optimisation problem.

If the iterative procedure is not adequately regularised, this technique may experience over-fitting. In some loss functions, such as quadratic loss, if the model h_m accurately fits the pseudo-residuals, the pseudo-Residuals will reach zero in the following iteration, causing the process to end prematurely. Multiple regularisation factors are taken into account to govern the additive procedure of gradient boosting. To regularise gradient boosting, a common approach is to apply shrinkage, which reduces the size of each gradient descent step $F_m(X) = F_{m-1}(X) + \nu p_m h_m(X)$ with $\nu = (0,1]$. The variable v is typically assigned a value of 0.1. Furthermore, further regularisation may be achieved by constraining the complexity of the trained models. When it comes to decision trees, we have the option to restrict the depth of the trees or specify the minimum number of occurrences required to divide a node. In contrast to random forest, gradient boosting sets default settings for these parameters that significantly restrict the trees' ability to represent complex relationships (e.g. the depth is often restricted to about 3-5). Additionally, the various implementations of gradient boosting use a set of settings that introduce randomness to the basic learners. This randomisation, such as random subsampling without replacement, may enhance the ensemble's ability to generalise.

- The characteristics that were ultimately investigated for gradient boosting are:
- The learning rate (*learning_rate*) or shrinkage ν.
- The maximum depth of the three (max_depth) : The same significance as found in the trees produced in a random forest.
- The subsampling rate (*subsample*) for the size of the random samples. Unlike random forest, this
- procedure is often conducted without replacement.
- The number of features to consider when looking for the best split (max_features): as in random forest.
- The minimum number of samples required to split an internal node (min_*samples*_*split*): as in random forest.

Algorithm 1 Gradient Boosting Classifier Implementation 1: Import Necessary Libraries:
2: - from sklearn.ensemble import GradientBoostingClassifier
3: - from sklearn.metrics import confusion matrix, precision score, f1 score, accuracy score, recall score
4: - import matplotlib.pyplot as plt 5: - import seaborn as sns

6: Create and Train a Gradient Boosting Classifier:
7: - Instantiate GradientBoostingClassifier with specified parameters (e.g., n estimators, max depth)
8: - Fit the classifier on the training data
9: Make Predictions:
10: - Predict labels for the test data
11: Calculate Metrics and Plot Confusion Matrix: 12: - Calculate confusion matrix
13: - Plot the confusion matrix using seaborn

3.3 Navis Bayes

The Naive Bayes method determines the likelihood that an item with certain qualities belongs to a specific group or category. Simply described, it's a probability-based classifier. The Naive Bayes technique is called after its assumption that one attribute is more prevalent than others. To identify fake accounts, consider factors such as language, location, and time of posting. These features, whether independent or dependent on other factors, raise the likelihood of a misleading profile.

It is mostly used for text classification using a large training set.

The Naive Bayes Classifier is a simple and efficient classification method accessible today. It helps create fast and accurate machine learning models.

A probabilistic classifier predicts the probability of an object's occurrence.

$$P\left(\frac{Y}{X}\right) = \frac{P\left(\frac{X}{Y}\right).P\left(Y\right)}{P(X)}$$

This method is great for categorising fresh data despite little data. It can handle datasets as few as 10 or 20 elements.

IV.METHODOLOGY

The main aim of this research is to identify instances of fraudulent activity in financial transactions. The approach we propose is an ensemble model that combines three machine learning models: random forest, gradient boosting, and Naive Bayes.

Data set Description

This dataset provides a synthetic representation of mobile money transactions that has been methodically created to mimic the intricacies of real-world financial activity while also include fraudulent behaviors for research purposes. This dataset, derived from a simulator called PaySim, uses aggregated data from genuine financial logs of a mobile money service and intends to address a vacuum in publicly accessible financial datasets for fraud detection research.

https://www.kaggle.com/datasets/sriharshaeedala/financial-fraud-detection-dataset

Pre-processing:

Data preparation involves the identification of missing data and the determination of whether to impute or remove them. Applying normalisation or standardisation to numerical features guarantees that the scales are consistent. In order to train a model, it may be necessary to convert category variables into numerical values. Feature engineering involves the creation or transformation of features in order to enhance the accuracy of model predictions.

Train test split: Once all the necessary pre-processing has been completed, the dataset is divided into training and testing sets according to the split ratio specified by the user. Subsequently, the divided train data will be used for model training, while the test data will be employed for model testing.

Model Building:

During the model-building phase, we will use an Ensemble learning approach that integrates many base models, including Random Forest, Gradient Boosting, and Naive Bayes models. To enhance overall predictive accuracy and robustness, we will use techniques such as bagging, boosting, and stacking to amalgamate the predictions of these models. The purpose of our Ensemble model is to provide a robust tool for early detection of fraudulent activities by integrating the knowledge and findings from several models.

Hyperparameter Tuning:

Hyperparameter tuning improves machine learning model hyperparameters with approaches such as Grid Search CV. This includes thoroughly evaluating several hyperparameter values to find which combination yields the best performance for each model. By fine-tuning these parameters, the models may achieve more accuracy and generalisation on previously unknown data, hence improving their overall prediction ability. Train the network:

The proposed ensemble learning model, along with established models like random forest, Gradient Boosting, and Naive Bayes, is trained using the training data. The performance of the recommended

model is assessed and contrasted with that of current models.

Performance metrics

Accuracy: The frequency with which a classifier generates accurate predictions is a direct expression that defines accuracy. The ratio of accurate predictions to the total number of predictions generated by the model can be used to define accuracy.

$$Accuracy = \frac{TP + TN}{S}$$

Precision: Precision is the ratio of the number of correctly classified instances to the total number of instances that have been classified.

$$\Pr ecision = \frac{TP\ TP}{+FP}$$

Recall: The ratio of correct positive numbers to the total number of true and false negatives.

$$\mathrm{Re}call = \frac{TP\ TP}{+FN}$$

F1-Score: The F1 score is determined by calculating the harmonic mean of the recall and accuracy values.

$$F1 = \frac{2 * \Pr ecision * \mathrm{Re}call}{\Pr ecision + \mathrm{Re}call}$$

V.RESULTS

Confusion Matrix

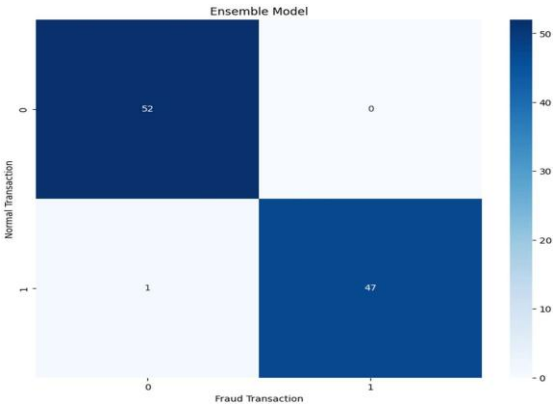


Figure 2 Ensemble Model

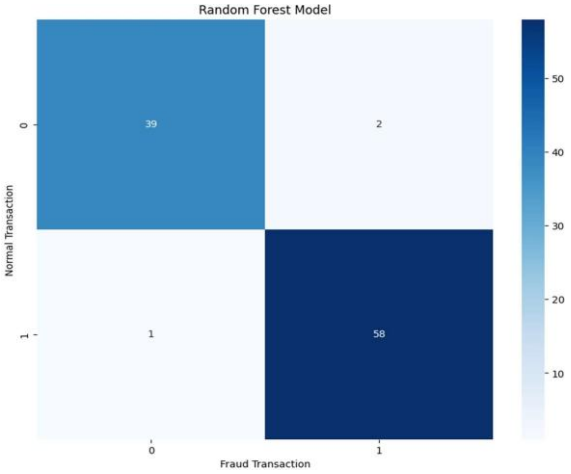


Figure 3 Random Forest

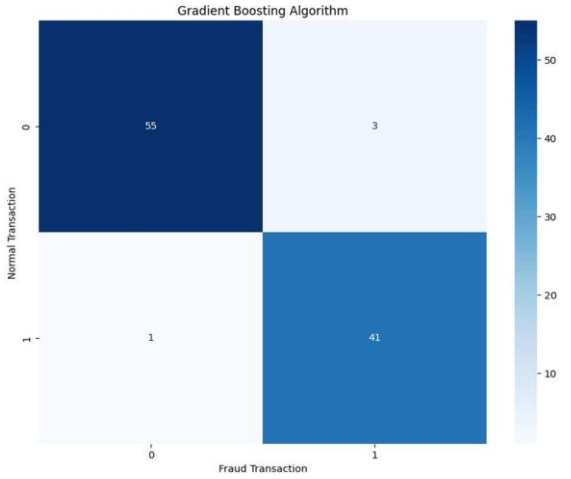
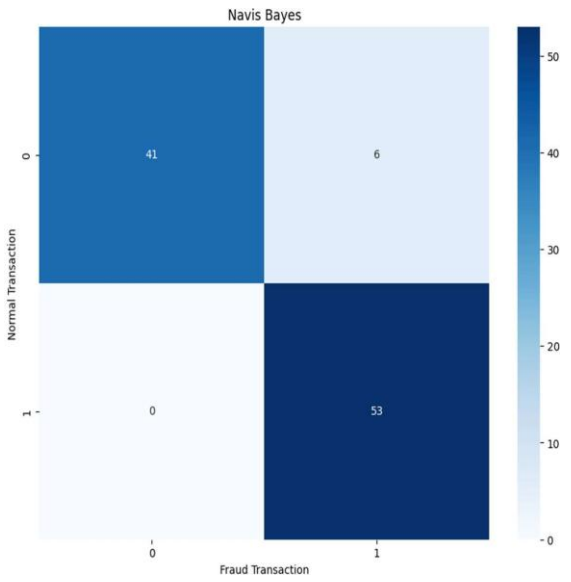


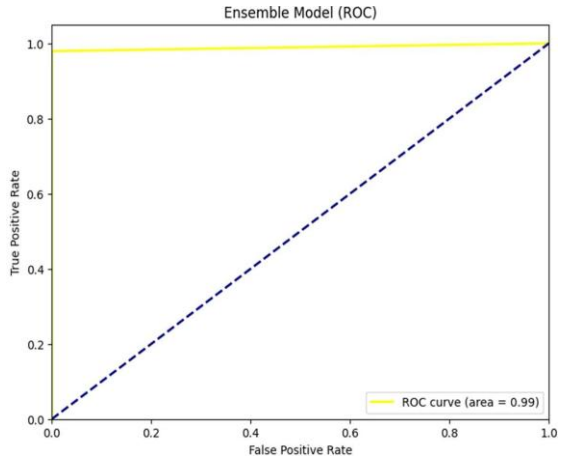
Figure 4 Gradient Boosting Algorithm

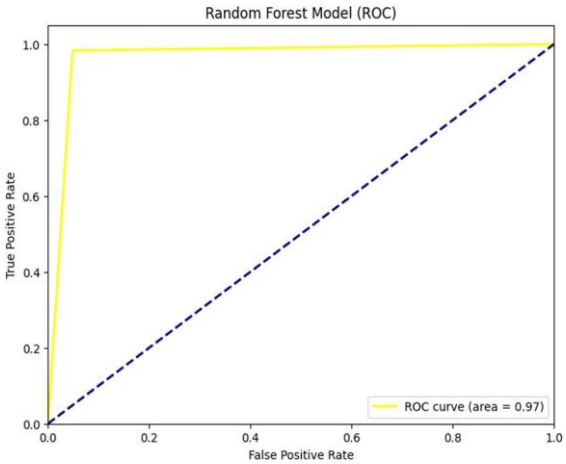


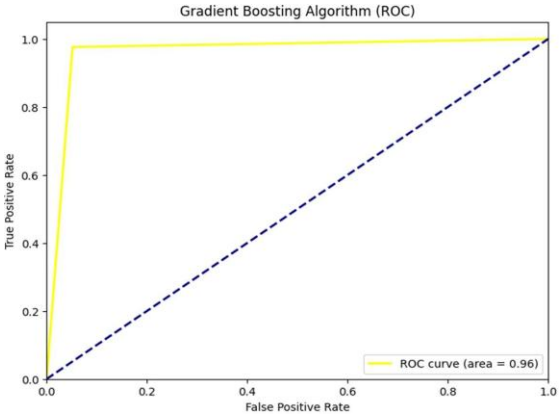
The confusion matrices examine the ability of three models (Ensemble Model, Random Forest Model, and Gradient Boosting Algorithm) to identify regular and fraudulent transactions. The Ensemble Model

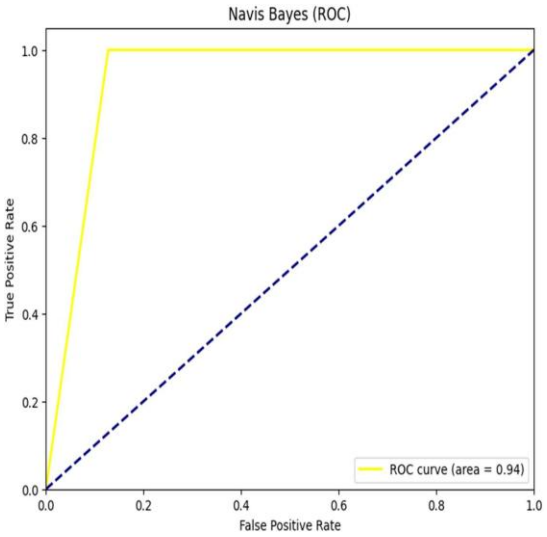
yields 52 true negatives, 47 true positives, one false negative, and zero false positives, demonstrating great accuracy and precision. The Random Forest Model produces 39 true negatives, 58 true positives, one false negative, and two false positives, indicating strong sensitivity but somewhat poorer specificity. The Gradient Boosting Algorithm returns 55 true negatives, 41 true positives, one false negative, and three false positives, indicating high accuracy but a few extra erroneous positives. The Navis Bayes returns 41 true negatives, 53 true positives, six false negatives, and zero false positives, indicating high accuracy. Overall, the Ensemble Model is the best at recognising real positives while reducing false positives, followed by the Random Forest and Gradient Boosting models.

ROC Curves









A comparative comparison of multiple machine learning models based on their ROC (Receiver Operating Characteristic) curves shows that the ensemble model beats the individual methods with a ROC of 0.99. This shows that the model outperforms the others in terms of class distinction. The Random Forest model follows with a ROC of 0.97, exhibiting good accuracy but somewhat lower than the ensemble technique. Gradient Boosting Algorithm performs well, with a ROC of 0.96, just slightly below Random Forest. Naive Bayes, although still successful, has a lower ROC of 0.94. The ensemble model's higher ROC implies that integrating many models might exploit their strengths, resulting in better predictive performance and resilience than any single model in isolation.

Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
Navis Bayes	0.94	0.94	0.94	0.93
Gradient Boosting Algorithm	0.96	0.961	0.96	0.960

Random Forest Model	0.97	0.970	0.97	0.969
Ensemble Model	0.99	0.990	0.99	0.98

When comparing the performance measures of several machine learning models, the suggested ensemble model has the greatest accuracy of 0.99, precision of 0.990, recall of 0.99, and F1-score of 0.98. This demonstrates an extraordinary capacity to accurately detect both positive and negative cases with few false positives and false negatives. The Random Forest model follows, with remarkable metrics such as 0.97 accuracy, 0.970 precision, 0.97 recall, and 0.969 F1-score. The Gradient Boosting Algorithm works well, with an accuracy of 0.96, precision of 0.961, recall of 0.96, and F1-score of 0.960. While the Naive Bayes model is successful, it trails behind the other models in terms of accuracy, precision, recall, and F1-score, with 0.94, 0.94, and 0.93 respectively. The ensemble model's better performance across all major criteria demonstrates its durability and efficacy in utilizing the capabilities of numerous algorithms to produce optimal overall prediction performance.

VI.CONCLUSION

In conclusion, the proposed ensemble model outperforms the previous machine learning models tested for identifying normal and fraudulent transactions. After conducting a thorough examination, the ensemble model attained the best accuracy of 0.99, precision of 0.990, recall of 0.99, and F1-score of 0.98. These measurements demonstrate its excellent ability to accurately detect both positive and negative situations while reducing mistakes. The confusion matrix study demonstrates the ensemble model's accuracy and recall, with 52 true negatives, 47 true positives, one erroneous negative, and no false positives. This shows that the ensemble model is quite good in reducing Type I (false positives) and Type II (false negatives) mistakes.

When ROC curves are compared, the ensemble model once again surpasses the other models, with a ROC of 0.99 indicating near-perfect class classification. This is much greater than the Random Forest model, which has an ROC of 0.97, indicating good performance but falling short of the ensemble model's capabilities. The Gradient Boosting Algorithm (ROC=0.96) and the Naive Bayes model (ROC=0.94) demonstrate the ensemble model's higher class differentiation and overall efficacy.

The Random Forest model, despite achieving

commendable metrics such as an accuracy of 0.97, a precision of 0.970, a recall of 0.97, and an F1-score of 0.969, as well as a confusion matrix containing 39 true negatives, 58 true positives, one false negative, and two false positives, falls short of the ensemble model's sensitivity and specificity. The Gradient Boosting Algorithm, despite having a high accuracy of 0.96, precision of 0.961, recall of 0.96, and an F1-score of 0.960, as well as a confusion matrix indicating 55 true negatives, 41 true positives, one false negative, and three false positives, falls short in comparison due to its slightly higher rate of false positives. The Naive Bayes model performs well, with an accuracy of 0.94, precision of 0.94, recall of 0.94, and F1-score of 0.93, but falls below the other models. Its confusion matrix, which has 41 true negatives, 53 true positives, six false negatives, and no false positives, shows a larger amount of erroneous negatives, reducing its capacity to effectively detect fraudulent transactions.

Overall, the ensemble model's capacity to combine the capabilities of many methods leads to greatly improved prediction performance and resilience. Its strong metrics across all major performance measures demonstrate its usefulness as the best model for distinguishing between normal and fraudulent transactions, making it a highly dependable and efficient alternative for this crucial job.

ACKNOWLEDGMENT

We thankfully acknowledge our institution for infrastructure and ethical support. Additionally, we respect all scientists and researchers for their dedication and contribution to enriching us. The authors have no conflict of interest.

REFERENCES

[1] Abbasi, A., Albrecht, C., Vance, A., & Hansen, J. (2012). MetaFraud: A Meta-Learning Framework for Detecting Financial Fraud. *MIS Quarterly*, *36*(4), 1293–1327. https://doi.org/10.2307/41703508

[2] Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, *6*, 100163. https://doi.org/https://doi.org/10.1016/j.dajour.2023.100163

[3] Albizri, A., Appelbaum, D., & Rizzotto, N. (2019). Evaluation of financial statements fraud detection research: a multi-disciplinary analysis. *International Journal of Disclosure and Governance*, *16*(4), 206–241. https://doi.org/10.1057/s41310-019-00067-9

[4] An, B., & Suh, Y. (2020). Identifying financial statement fraud with decision rules obtained from Modified Random Forest. *Data Technologies and Applications*, *54*(2), 235–255. https://doi.org/10.1108/DTA-11-2019-0208

[5] Arisa, N. A., Othmanb, R., Mohd, M. A., Bukhoria, Arifa, S. M. M., & Malek, and M. A. A. (2018). Detecting accounting anomalies using Benford's law: evidence from the Malaysian public sector. *MANAGEMENT & Accounting Review*, *16*(2), 73–99.

[6] Baesens, B., Höppner, S., Ortner, I., & Verdonck, T. (2021). robROSE: A robust approach for dealing with imbalanced data in fraud detection. *Statistical Methods & Applications*, *30*(3), 841–861. https://doi.org/10.1007/s10260-021-00573-7

[7] Bao, Y., Ke, B., Li, B., Yu, Y. J., & Zhang, J. (2020). Detecting Accounting Fraud in Publicly Traded U.S. Firms Using a Machine Learning Approach. *Journal of Accounting Research*, *58*(1), 199–235. https://doi.org/10.1111/1475-679X.12292

[8] Beneish, M. D., & Vorst, P. (2022). The Cost of Fraud Prediction Errors. *Accounting Review*, *97*(6), 91–121. https://doi.org/10.2308/TAR-2020-0068

[9] Cao, D. F., & Liu, B. C. (2019). SVM Model for Financial Fraud Detection. *Dongbei Daxue Xuebao/Journal of Northeastern University*, *40*(2). https://doi.org/10.12068/j.issn.1005-3026.2019.02.027

[10] Çığşar, B., & Ünal, D. (2019). Comparison of Data Mining Classification Algorithms Determining the Default Risk. *Scientific Programming*, *2019*, 8706505. https://doi.org/10.1155/2019/8706505

[11] du Jardin, P. (2016). A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research*, *254*(1), 236–252. https://doi.org/https://doi.org/10.1016/j.ejor.2016.03.008

[12] Du, X., Li, W., Ruan, S., & Li, L. (2020). CUS-

heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection. *Applied Soft Computing*, *97*, 106758. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106758

[13] Eweoya, I. O., Adebiyi, A. A., Azeta, A. A., & Azeta, A. E. (2019). Fraud prediction in bank loan administration using decision tree. *Journal of Physics: Conference Series*, *1299*(1). https://doi.org/10.1088/1742-6596/1299/1/012037

[14] Faraji, Z. (2022). A Review of Machine Learning Applications for Credit Card Fraud Detection with A Case Study. *SEISENSE Journal of Management*, *5*(1), 49–59. https://doi.org/10.33215/sjom.v5i1.770

[15] Guleria, P., Ahmed, S., Alhumam, A., & Srinivasu, P. N. (2022). Empirical Study on Classifiers for Earlier Prediction of COVID-19 Infection Cure and Death Rate in the Indian States. *Healthcare (Switzerland)*, *10*(1). https://doi.org/10.3390/healthcare10010085

[16] Handoko, B. L., & Natasya. (2019). Fraud diamond model for fraudulent financial statement detection. *International Journal of Recent Technology and Engineering*, *8*(3), 6865–6872. https://doi.org/10.35940/ijrte.C5838.098319

[17] Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Systems with Applications*, *193*, 116429. https://doi.org/https://doi.org/10.1016/j.eswa.2021.116429

[18] Hosaka, T. (2019). Bankruptcy prediction using imaged financial ratios and convolutional neural networks. *Expert Systems with Applications*, *117*, 287–299. https://doi.org/https://doi.org/10.1016/j.eswa.2018.09.039

[19] Ijaz, M. F., Alfian, G., Syafrudin, M., & Rhee, J. (2018). Hybrid Prediction Model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, Synthetic Minority Over-Sampling Technique (SMOTE), and random forest. *Applied Sciences (Switzerland)*, *8*(8). https://doi.org/10.3390/app8081325

[20] Jan, C. L. (2021). Detection of financial statement fraud using deep learning for sustainable development of capital markets under information asymmetry. *Sustainability (Switzerland)*, *13*(17). https://doi.org/10.3390/su13179879

[21] Kabir, E., Guikema, S., & Kane, B. (2018). Statistical modeling of tree failures during storms. *Reliability Engineering & System Safety*, *177*, 68–79. https://doi.org/https://doi.org/10.1016/j.ress.2018.04.026

[22] Kamal, M. E. M., Salleh, M. F. M., & Ahmad, A. (2016). Detecting financial statement fraud by Malaysian public listed companies: The reliability of the Beneish M-Score model. *Jurnal Pengurusan*, *46*, 23–32. https://doi.org/10.17576/pengurusan-2016-46-03

[23] Kumar, M. S., Soundarya, V., Kavitha, S., Keerthika, E. S., & Aswini, E. (2019). Credit Card Fraud Detection Using Random Forest Algorithm. *2019 Proceedings of the 3rd International Conference on Computing and Communications Technologies, ICCCT 2019*, 149–153. https://doi.org/10.1109/ICCCT2.2019.8824930

[24] Liang, D., Tsai, C.-F., Lu, H.-Y. (Richard), & Chang, L.-S. (2020). Combining corporate governance indicators with stacking ensembles for financial distress prediction. *Journal of Business Research*, *120*, 137–146. https://doi.org/https://doi.org/10.1016/j.jbusres.2020.07.052

[25] Lin, B., & Bai, R. (2022). Machine learning approaches for explaining determinants of the debt financing in heavy- polluting enterprises. *Finance Research Letters*, *44*, 102094. https://doi.org/https://doi.org/10.1016/j.frl.2021.102094

[26] Maniatis, A. (2022). Detecting the probability of financial fraud due to earnings manipulation in companies listed in Athens Stock Exchange Market. *Journal of Financial Crime*, *29*(2), 603–619. https://doi.org/10.1108/JFC-04-2021-0083

[27] Ofori, E. (2016). Detecting Corporate Financial Fraud Using Modified Altman Z-Score and Beneish M-Score. The Case of Enron Corp. In *Research Journal of Finance and Accounting* (Vol. 7, Issue 4, pp. 59–65).

[28] Omar, N., Johari, Z. 'Amirah, & Smith, M. (2017). Predicting fraudulent financial reporting using artificial neural network.

Journal of Financial Crime, *24*(2), 362–387. https://doi.org/10.1108/JFC-11-2015-0061

[29] Pisula, T. (2020). An Ensemble Classifier-Based Scoring Model for Predicting Bankruptcy of Polish Companies in the Podkarpackie Voivodeship. *Journal of Risk and Financial Management*, *13*(2). https://doi.org/10.3390/jrfm13020037

[30] R, P. T. (2015). A Comparative Study on Decision Tree and Random Forest Using R Tool. *Ijarcce*, *January 2015*, 196–199. https://doi.org/10.17148/ijarcce.2015.4142

[31] Rathore, A. S., Kumar, A., Tomar, D., Goyal, V., Sarda, K., & Vij, D. (2021). Credit Card Fraud Detection using Machine Learning. *Proceedings of the 2021 10th International Conference on System Modeling and Advancement in Research Trends, SMART 2021*, 167–171. https://doi.org/10.1109/SMART52563.2021.9676262

[32] Ratmono, D., Darsono, D., & Cahyonowati, N. (2020). Financial Statement Fraud Detection With Beneish M-Score and Dechow F-Score Model: An Empirical Analysis of Fraud Pentagon Theory in Indonesia. *International Journal of Financial Research*, *11*(6), 154. https://doi.org/10.5430/ijfr.v11n6p154

[33] Ruan, S., Sun, X., Yao, R., & Li, W. (2021). Deep Learning Based on Hierarchical Self-Attention for Finance Distress Prediction Incorporating Text. *Computational Intelligence and Neuroscience*, *2021*, 1165296. https://doi.org/10.1155/2021/1165296

[34] Sohl, J. E., & Venkatachalam, A. R. (1995). A neural network approach to forecasting model selection. *Information & Management*, *29*(6), 297–303. https://doi.org/https://doi.org/10.1016/0378-7206(95)00033-4

[35] Tasnim, A., Saiduzzaman, M., Rahman, M. A., Akhter, J., & Rahaman, A. S. M. M. (2022). Performance Evaluation of Multiple Classifiers for Predicting Fake News. *Journal of Computer and Communications*, *10*(09), 1–21. https://doi.org/10.4236/jcc.2022.109001

[36] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, *57*, 47–66. https://doi.org/https://doi.org/10.1016/j.cose.2015.09.005

[37] Xiong, T., Ma, Z., Li, Z., & Dai, J. (2022). The analysis of influence mechanism for internet financial fraud identification and user behavior based on machine learning approaches. *International Journal of System Assurance Engineering and Management*, *13*(3), 996–1007. https://doi.org/10.1007/s13198-021-01181-0

[38] Ye, H., Xiang, L., & Gan, Y. (2019). Detecting Financial Statement Fraud Using Random Forest with SMOTE. *IOP Conference Series: Materials Science and Engineering*, *612*(5). https://doi.org/10.1088/1757-899X/612/5/052051

[39] Yusrianti, H., Ghozali, I., Yuyetta, E., Aryanto, & Meirawati, E. (2020). Financial statement fraud risk factors of fraud triangle: Evidence from Indonesia. *International Journal of Financial Research*, *11*(4), 36–51. https://doi.org/10.5430/ijfr.v11n4p36