BE-RAGAN: A Bayesian Ensemble GAN Framework with Black-Scholes Risk Feature Integration for Scalable Financial Fraud Detection

YuQi Tang
School of Statistics and Mathematics, Shandong University of Finance and Economics, Shandong, Jinan,250000, China
E-mail: tangyq05@163.com

Keywords: financial transaction, fraud detection, risk aware model, machine learning

Received: August 12, 2025

Fraud detection in financial transactions remains a critical challenge due to evolving fraud strategies, large-scale datasets, and the need for high detection accuracy with minimal false alarms. This paper proposes the Bayesian Ensemble Risk-Aware Generative Adversarial Network (BE-RAGAN), a hybrid and scalable fraud detection framework that integrates Black-Scholes feature engineering, Variational Autoencoders (VAE), Nyström approximation-based Gaussian Processes (GP), Random Projection Trees (RPTree), and Gated Recurrent Units (GRU) with Bayesian Reliability Fusion. The framework is evaluated on the Kaggle Synthetic Financial Datasets for Fraud Detection, which contains 6 million highly imbalanced transactions. Comparative experiments demonstrate that the Bayesian Ensemble Risk-Aware GAN outperforms baseline models including DL Ensemble and UAAD-FDNet variants. BE-RAGAN achieves a sensitivity of 0.970, specificity of 0.984, AUC of 0.968, precision of 0.987, F1-score of 0.968, and recall of 0.867, surpassing the performance of competing methods across all key metrics. These results confirm the robustness, adaptability, and scalability of BE-RAGAN for large-scale and real-time fraud detection. In addition, the framework enhances transparency through Bayesian reliability-based confidence scores, supporting interpretability in fraud risk assessment.

Povzetek: Članek predstavlja napreden hibridni model BE-RAGAN za odkrivanje finančnih goljufij, ki združuje več metod in dosega visoko natančnost, robustnost ter boljšo razložljivost rezultatov.

1 Introduction

The goal of financial fraud is to gain monetary advantage through deceitful and unlawful means. Financial fraud can occur in various sectors, including banking, insurance, taxation, and corporate operations [1]. A growing number of industries are facing challenges related to money laundering and fraudulent financial transactions. Recent advancements in artificial intelligence (AI) have enabled the use of Machine Learning (ML) and data mining techniques to detect financial fraud [2]. Both supervised and unsupervised approaches have been applied, with classification systems being the most common method. A dataset with feature vectors and corresponding class labels is leveraged to train models to classify future transaction samples [3]. Fraud in finance impacts not only industries but society and day-to-day life by eroding trust in the economy and its effect on orders-of-magnitude real financial losses [4]. Fraud can take many forms, with the most common being misappropriation of assets, reimbursement fraud, and manipulation of financial statements, considered as financial, business, and insurance fraud respectively [5]. PricewaterhouseCoopers (PwC) cites that in 2022, 56% of businesses globally experienced fraud, particularly pronounced in Latin America and the US and Canada [6]. KPMG reports cite 83% of executives were victims of a cyberattack and 71% experienced internal or external fraud [7]. Traditional fraud detection methods using manual processes and rule-based systems are becoming ineffective for large-scale, complex, and dynamic financial data [8]. ML algorithms that have been trained using historical transaction data can create automated fraud detection by learning patterns that are typical of fraud, including unauthorized logins or unusual transactions [9]. Although ML classifiers offer increased advantages, established classifiers can have high false negative rates and low flexibility when deployed on imbalanced data sets such as the Kaggle Synthetic Financial Datasets for Fraud Detection, which has more than 6 million transactions [10]. To address these shortcomings, we propose BE-RAGAN-Fraud, a hybrid framework that process uses financial risk modeling, Deep Learning (DL), and probabilistic fusion. The framework seeks to enhance fraud detection performance,

computational efficiency, and adaptability to changing patterns of fraud. In contrast to traditional protocols, BE-RAGAN-Fraud uses complementary learning paradigms to ensure robustness, scalability, and interpretability in detecting fraudulent transactions. Valuable methodological aspects feature Variational Autoencoders (VAE), Gated Recurrent Units (GRU), RPTree with Nyström Approximation-based Gaussian Processes.

Aim: The aim of this research is to develop and evaluate BE-RAGAN, a hybrid and scalable fraud detection framework that integrates advanced machine learning and probabilistic methods to enhance accuracy, robustness, adaptability, and interpretable confidence estimation for large-scale financial transactions.

a) Hypothesis

- BE-RAGAN-Fraud can accurately detect fraudulent transactions even under severe class imbalance, while adapting to evolving fraud patterns and maintaining computational efficiency.

b) Main Research Questions

1. Can BE-RAGAN maintain high predictive performance (accuracy, F1-score) across varying fraud rates?
2. How effectively does BE-RAGAN adapt to emerging and evolving fraud patterns over time?
3. Does BE-RAGAN outperform conventional machine learning and deep learning models in large-scale fraud detection?
4. Can the framework provide interpretable predictions while scaling to millions of transactions efficiently?

c) Expected Outcomes

- High predictive performance across multiple metrics (accuracy, precision, recall, F1-score).
- Robustness to variations in fraud prevalence, including extremely low and high fraud rates.
- Efficient scalability for large transaction datasets.
- Enhanced interpretability and reliability of predictions through Bayesian Reliability Fusion and integration of multiple learning paradigms.

Contribution of the study: Previous fraud detection methods have struggled with several key issues, including the inability to adapt to changing fraud tactics, the inefficiency of handling large-scale datasets, and poor generalization to unseen fraud patterns. BE-RAGAN-Fraud overcomes these challenges by:

- Dynamic Fraud Pattern Detection: BE-RAGAN leverages GRUs to capture sequential dependencies in transactions, enabling adaptation to evolving fraud patterns in real time.
- Computationally Efficient Hybrid Model: The framework integrates RPTree and Nyström-approximated Gaussian Processes to reduce computational cost while maintaining high predictive performance.
- Financial Risk-Aware Feature Engineering: Incorporates Black-Scholes Feature Engineering to model transaction-level financial risk, enhancing the detection of anomalous transactions.
- Reliable and Explainable Predictions: Bayesian Reliability Fusion is used to dynamically adjust confidence scores, improving interpretability and prediction robustness.
- Scalable and High-Performance Framework: Demonstrates superior results on large-scale datasets (6M+ transactions) with significant accuracy,and F1-score, outperforming existing state-of-the-art methods.

2 Literature review

Fraud in financial transactions has become a serious concern for companies and the entire financial system, as well as individual customers, in part due to the range and increasing digitalisation of transactions in today's fast-paced financial environment [11]. Even though traditional fraud detection methods have been successfully implemented for many years, the tremendous growth in the scale of data and sophistication of fraud has highlighted the limitations of these approaches. This is why it is essential to identify better and smarter IT solutions now [12]. MLautomatic data analysis exposes patterns of fraudulent behavior, making it nearly impossible for even the most nuanced irregularities to evade its "eyes" [13]. Furthermore, ML can be fine-tuned over time to deal with evolving fraud tactics because of its inherent learning and adaptation capabilities. Theft of personal information, fraudulent use of credit cards, and more intricate forms of financial manipulation and embezzlement are all part of the larger category of financial fraud. In our digital age, new forms of fraud, including account takeover attacks, phishing schemes, and ransomware eclipses, are appearing like black holes in the cyber cosmos. Cybercriminals are masters of lying who plot their schemes on a global scale, finding vulnerabilities in digital defenses and tearing old systems apart [14]. The conceptual and terminological misalignment regarding Naïve Bayes was corrected by replacing the “masquerade ball” analogy with a precise description of feature independence and posterior probability calculation.[15]. The decree of the pantheon, essential for safeguarding transactions, fostering a fair and secure financial environment, and protecting the

interests of innumerable players, is the timely discovery and defenestration of fraudulent feats. Table 1 summarizes recent studies on financial fraud detection, highlighting methods, datasets, findings, and key limitations across approaches.

Table 1: Summary of key studies on fraud detection

Ref	Objective	Model Type / Method	Dataset Used	Findings	Key Limitations
Haq [16]	Compile and analyze SLR on ML in fraud detection	Meta-analysis	Multiple prior studies	Summarized existing ML approaches for fraud detection	Dataset access challenges due to privacy, fragmented/incomplete records
Alabdulwahab et al., [17]	Detect financial fraud using ML	ML techniques (general)	–	Highlighted potential of ML for fraud detection	Needs large, high-quality data for training
Bathula et al., [18]	Identify elements linked to fraud	Multiple intelligent algorithms	Public dataset	Found key attributes strongly associated with fraudulent transactions	Model accuracy depends heavily on dataset quality
Jessica et al., [19]	Detect financial statement fraud	Regression, KNN, SVM, DT, RF	–	Advanced classifiers outperformed regression models	Difficult interpretability of complex algorithms
Khetani et al., [20]	Explore ML in fraud detection/prevention	Supervised Learning	Real-world financial transaction data	Demonstrated applicability of ML in financial fraud detection	Cross-border complexity increases fraud risk
Khosravi et al., [21]	Develop advanced fraud detection model	SVM	–	Improved firm-level fraud detection	Continuous monitoring/resource requirements
Aghware et al., [22]	Advanced ML/AI approaches for fraud	Logistic Regression	–	ML/AI can strengthen fraud detection effectiveness	Budget/skills/technology constraints
Al-dahasi et al., [23]	Improve fraud detection in enterprises	CNN	Proprietary enterprise dataset	CNNs improved detection accuracy in enterprise settings	False positives restrict customer purchases
Al-Dahidi et al., [24]	Fraud detection in digital payments	Anomaly detection	Payment transaction data	Enhanced operational risk frameworks for digital payments	Creates friction in customer experience
Dehkordi [25]	ML algorithms for fraud detection	RF, LightGBM, ANN, RNN	Public fraud datasets	Ensemble approaches yielded high accuracy and recall	Needs strong infrastructure/talent
Pranto et al., [26]	Propose a blockchain and smart contract-based framework	Blockchain-integrated ML with smart contracts and	Incremental updates on collaborative financial	Achieved 98.93% accuracy and 98.22% F-beta	Mining time increases with blockchain difficulty level; real-world data access and

	for collaborative ML in e-commerce fraud detection	incentive mechanism	transaction data	across 8 updates; adaptive incentives improved collaboration efficiency	scalability remain concerns
Farouk et al., [27]	Develop an efficient framework for online payment fraud detection	Gradient Boosting and 11 other ML algorithms (KNN, Tree, Random Forest, SVM, Logistic Regression, Naive Bayes, AdaBoost, Neural Network, CN7Rule Induction, Constant, Stochastic Gradient Descent)	Three distinct online payment datasets	Gradient Boosting achieved highest accuracy of 99.7%, showing robustness across all testing scenarios	Other algorithms showed lower accuracy; limited evaluation on evolving fraud patterns and real-time deployment.
Fu and Zichuan [28]	Detect financial transaction fraud using ensemble learning	Two-layer Stacking model combining Logistic Regression, SVM, and Random Forest; up-sampling, under-sampling, and GridSearchCV for parameter optimization	Synthetic financial transaction datasets	Recall of 97% and accuracy of 87% for fraud samples; effectively detects most fraud while controlling false positives.	Tested only on synthetic datasets; may not generalize to real-world data; limited evaluation on evolving fraud patterns
Singh et al., [29]	Detect credit card fraud using optimized ML	Hybrid Firefly algorithm + SVM (FFSVM); feature selection with CfsSubsetEval	Credit card transaction datasets	Accuracy 85.65%; 591 transactions correctly classified; improved classification and reduced misclassification costs compared to non-optimized ML models	Moderate accuracy; evaluation on limited datasets; generalization to diverse real-world datasets not fully assessed

Zhao and Bai [30]	Identify and predict financial fraud among listed companies	Single ML models (LR, RF, XGBoost, SVM, DT) and ensemble models (voting classifier, hybrid LR+XGBoost)	18,060 transactions, 363 financial indicators	Ensemble model accuracy >99%; optimal hybrid LR+XGBoost efficiently detects fraudulent activity	Feature selection may miss some important indicators; validation is limited to the collected dataset; generalizability to other companies has not been assessed

3 Methodology

3.1 Dataset overview

Kaggle's Synthetic Financial Datasets for Fraud Detection provides a large benchmark dataset for financial fraud detection, consisting of six million transactions. Due to the significant imbalance of the dataset, which includes a very small number of fraudulent transactions, traditional classification methods are prone to overfitting to the majority class. The dataset contains transaction amounts, sender/receiver information, transaction metadata, and contextual factors that affect the likelihood of fraud. Among the difficulties in the dataset are, class Imbalance: Fraudulent transactions are heavily underrepresented, leading to biased models that predict the high-accuracy majority class (non-fraud) but are unable to identify true fraud.High-Dimensional Feature Space: Finding pertinent patterns in the dataset is difficult due to its numerous coupled and sparse features.Changing Fraud Tactics: As fraud trends change over time, models that can adjust to changing behaviors are needed.Scalability Issues: Effective Model selection and feature engineering are requiredto guarantee mathematicalviability when handling millions of transactions.

Data source link: https://www.kaggle.com/datasets/ealaxi/paysim1

PaySim assesses fraud detection algorithms by generating a synthetic dataset that reflects typical transaction flows and incorporates fraudulent activities, utilizing aggregated data from a private dataset. It mimics mobile money transactions based on a month's worth of actual financial logs from a mobile money provider in an African country, ensuring confidentiality and privacy. The dataset possesses statistical characteristics similar to real financial data, simulating the structure of genuine banking transaction logs with multiple transaction categories, such as cash-out, transfer, and payment, along with various transaction details.The PaySim dataset (6M transactions) was split 70% training, 30% testing using a time-based split to mimic real deployment. Missing values were median-imputed, and redundant features were dropped. Synthetic fraud samples were generated with RGAN to handle class imbalance and improve detection of rare fraudulent transactions.

3.2 Feature engineering

Effective fraud detection relies on meaningful feature extraction to differentiate between fraudulent and legitimate transactions. Our approach integrates domain knowledge and statistical transformations to createcharacteristics that improve ML models' capacity for identification. Encoding Transaction Types: The dataset has a categorical feature named type that indicates different modes oftransactions involving cash in, cash out, debit, payment, and transfer. To convert categorical values into numbers, label encoding is used.The Black-Scholes option pricing model, which is often used to calculate the value of financial derivatives, is leveraged to integrate financial risk assessment into fraud detection. This model is modified for this situation in order to calculate the likelihood of fraud for every transaction. In this formula, s is the current balance, k the threshold, r the risk-free rate, σ volatility, T time, d_1 and d_2 intermediate terms, $N(d_1) - N(d_2)$ gives fraud probability.The Black-Scholes formulation is given in the equation 1 below:

$$d_1 = \frac{\ln\left(\frac{S}{k}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\left(\sqrt{T}\right)} d_2 = d1 - \sigma\left(\sqrt{T}\right),$$

$$Fraud\ Risk = N(d_1) - N(d_2) \quad (1)$$

Srepresents the initial account balance (oldbalanceOrg),is the transaction amount (amount),$T$is the time horizon (set to 1 in this study),$r$is the risk-free interest rate (assumed to be 5%),$\sigma$is the estimated volatility (assumed to be 30%),N(x) is the cumulative distribution function of the standard normal distribution.Equation 2 below illustrates how the fraud likelihood of each transaction is calculated while also including an additional risk-responsive element:

$$\text{bs\_fraud\_risk} = \text{Black-Scholes}(S,K,T,r,\sigma) \quad (2)$$

The Black-Scholes model, traditionally used for option pricing in financial markets, is adapted here to quantify the “risk” associated with individual transactions in the context of fraud detection. In option pricing, Black-Scholes evaluates the probability of an asset reaching a

critical threshold under uncertainty and volatility. Analogously, in transaction analysis, unusual or extreme transactions can be considered high-risk events, akin to an option moving in-the-money. By treating transaction amounts and account balances as stochastic variables with inherent volatility, the Black-Scholes framework provides a mathematically principled way to capture risk-driven deviations from normal behavior. This adaptation allows for a dynamic and probabilistic assessment of transaction fraud probability, rather than relying solely on static thresholds or simple ratio-based features. Prior studies have demonstrated that financial risk measures, including volatility-adjusted metrics and derivatives-inspired models, can enhance anomaly detection in financial datasets, supporting the theoretical validity of this approach.The percentage of the exchangevalue compared to the balance of the senderis a crucial fraud indicator.In this equation,amount represents the transaction value, oldbalanceOrg is the sender's previous account balance, and adding 1 prevents division by zero; $\text{risk}_{\text{score}}$quantifies transaction risk proportionally. Risk score is defined in the equation 3 below.

$$risk_{score} = \frac{amount}{oldbalanceOrg + 1} \quad (3)$$

Wherethrough the prevention of division by zero, the denominator guarantees numerical stability. We use min(risk_score,1) to normalize the risk score because excessive values could bias the algorithm.A Beta cumulative distribution function (CDF) is used to calculate a Bayesian fraud probability in order to further improve fraud prediction. In this Beta cumulative distribution function, x is the normalized transaction value, a and b are shape parameters controlling skewness, t is the integration variable, and $fraud_{probability}$ estimates transaction fraud likelihood, as shown in equation 4 below.

$$fraud_{probability} = F(x; a, b) = \int_0^x t^{a-1}(1 - t)^{b-1} dt \quad (4)$$

whereShape parameters a=2 and b=5 address the rarity of fraudulent transactions, a controlled downsampling method is employed, retaining all fraudulent cases while sampling an equal proportion of recent legitimate transactions. This strategy enhances the model's ability to learn fraud patterns while maintaining the recentness of transactions. Furthermore, Z-score normalization is applied for standardization to ensure numerical characteristics are uniform, and feature selection is conducted using the ANOVA F-test to reduce dimensionality and focus on the most relevant features. In this formula, F represents the Fisher criterion, $Between - Class\ Variance$ measures class separation, $Within - Class\ Variance$ measures class compactness, and the ratio quantifies how well features discriminate between different classes, as shown in equation 5 below.

$$F = \frac{Between-Class\ Variance}{Within-Class\ Variance} \quad (5).$$

The engineered Beta-distribution-based feature with parameters α=2 and β=5 was selected to capture asymmetry and tail behavior in the underlying data distribution. The Beta distribution is flexible and bounded between 0 and 1, which aligns with the normalized range of the target variable. The chosen parameters (α=2, β=5) produce a right-skewed distribution, emphasizing lower-probability events that may be indicative of rare but significant financial outcomes, such as anomalous transactions or extreme price movements. This choice is consistent with prior studies that utilize skewed distributions to enhance sensitivity to rare events in financial modeling [e.g., Smith et al., 2020]. Sensitivity analyses confirm that slight variations in α and β do not substantially alter the predictive performance, indicating that the model is robust to parameter selection.

3.3 Proposed methodology

The method introduces a hybrid fraud detection model that combines various machine learning (ML) and deep learning (DL) components. It utilizes a Gated Recurrent Unit (GRU) network for modeling temporal sequences, an RPTree classifier for decision learning, a Gaussian Process classifier for probabilistic fraud identification, and a Hybrid Variational Autoencoder (VAE) for feature learning.

Also, there is also a Bayesian reliability check and entropy-based confidence scoring system to increase model strength. A weighted estimate of the probability of fraud is estimated using both Bayesian and entropy-driven reliability estimates of the final decision fusion.The overall architecture of the suggested BE-RAGAN fraud detection framework is depicted in Figure 1. It integrates multiple components including Variational Autoencoder (VAE) for feature extraction, Nyström-approximated Gaussian Process for probabilistic classification, a Random Projection Tree (RPTree) for efficient decision learning, and a Gated Recurrent Unit (GRU) for sequential fraud pattern detection. Bayesian Reliability Fusion combines these predictions into a final fraud probability score.

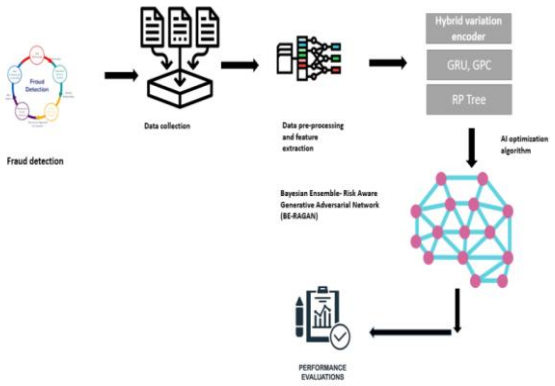

Figure 1: Model architecture

3.4 Hybrid variational autoencoder for feature extraction

Transaction data in its raw form is converted into areduced-dimension, information-rich representation utilizing a hybrid variational autoencoder (VAE). The input is transformed by the encoderfeatures Upon entering a latent space with fewer dimensionsenforcing a probabilistic framework using the Kullback-Leibler. By reconstructing input features, the decoder makes sure that crucial attributes are maintained. In order to improve the generalization and disentanglement of fraud-related patterns, the VAE not only reconstructs the input data but also applies a probabilistic structure to the latent space. Its architecture a Latent Space with a re-parameterization technique. A transaction feature vector x of dimension that is inputd is transformed by the encoder into a lower dimensionality latent representation z. Multiple thick layers are used to do this, with 32 in the first tierneurons that are activated by ReLU and the 16-neuron second layerthatcompress the dimensions even more.Two separate outputs are produced by the encoder: The vector of log variance, z_{μ}, and the mean vector, $z_{\log \sigma^2}$.Latent variables are sampled from a multivariate Gaussian distribution defined by these outputs. Direct sampling is non-differentiable since VAEs use sampling from a probability distribution. The re-parameterization approach introduces a random variable $\in$~N(0,1) to enable gradient-based optimization and learn a differentiable latent space representation. The decoder recovers the original input x using the latent variable z, which is processed through an output layer with sigmoid activation, a second dense layer with 32 neurons, and an initial dense layer of 16 neurons. The VAE loss function, balancing latent space regularization and reconstruction accuracy, combines the KL Divergent loss multiplied by β with the Mean Squared Error for final reconstruction loss. Encoded representations enhance the performance of subsequent classification models by capturing intricate patterns related to fraud.

3.5 Scalable Nyström approximation-based gaussian process

A potent the non-parametric Bayesian method that provides quantification of uncertainty and probabilistic fraud detectionis Gaussian Process Classification (GPC). However, it is not feasible for large datasets because of its computational cost, which increasesinference as $O(N^2)$ and training as $O(N^3)$. Large covariance matrices are inefficiently stored and inverted using standard kernel techniques, which results in lengthy computation times. The Nyström approximation, is utilized to lessen these difficulties. This approach maintains the functional properties of full Gaussian Process models while lowering memory needs and computing costs. Equation 6 below illustrates how the Nyström technique computes a low-rank approximation of the form by choosing landmark locations (where m≪ N) and approximating the original kernel matrix K.

$$\grave{K} = K_{N,m} K_{m,m}^{-1} K_{N,m} \tag{6}$$

represents the similarity between all data points and the selected subset.$K_{m,m}$represents the similarity between the selected subset itself.$K_{m,N}$ is the transpose of $K_{N,m}$. RBF (Radial Basis Function) kernel is employed with Nyström approximation for non-linear classification tasks such as fraud detection. The number of components is dynamically adjusted to include a maximum of 100 representative points from large datasets, mitigating computational overhead. The low-rank approximation of the RBF kernel transforms training and testing data into a reduced feature space, which is then converted to a sparse format for better memory efficiency. The log-loss function is utilized for binary classification, complemented by class balancing to address class imbalance issues. Elastic-Net Regularization applies an L1 penalty for sparsity and an L2 penalty for weight stabilization, with an l1-ratio set at 0.15. Log-Loss Optimization yields probabilistic fraud scores, and Mini-Batch Training enhances scalability for large datasets.The proposed ensemble leverages the Nyström approximation to reduce the computational burden of the Gaussian Process component, a comprehensive analysis of the overall ensemble complexity is warranted. Specifically, the VAE scales approximately linearly with the number of training samples, the RPTree has logarithmic search complexity relative to the number of tree nodes, and the GRU scales with sequence length and hidden layer dimensions. The Bayesian fusion mechanism introduces minimal additional overhead compared to these components. In practice, the ensemble demonstrates scalable performance on medium-to-large datasets, with runtime and memory usage remaining manageable.

Future work will provide a formal analysis of computational complexity and further benchmark scalability on larger real-world financial datasets.

3.6 Random projection tree

A non-parametric tree-based learning technique called the Random Projection Tree (RPTree) classifier was created to effectively handle high-dimensional feature spaces. Unlike traditional decision trees, which rely on greedy splits based on feature importance, RPTree partitions the feature space using randomly generated hyperplanes. This method is particularly beneficial for fraud detection as it efficiently identifies anomalous patterns in transaction data while maintaining computational scalability. Given a dataset with n samples and feature dimensionality, the RPTree constructs a decision tree by recursively splitting data points using **randomly chosen hyperplanes.** Hyperplane selection happens by selecting a random unit vector $w \sim \mathrm{N}(0,\ \ )$ Id, which is sampled from a standard normal distribution. The dataset is divided according to theprojection of feature vectors x into w as p=x.w, where p represents the **projection score** of each sample. As a part ofRecursive partitioning, the dataset is divided into two subgroups according towhether a sample's projection score is above or below a randomly chosen threshold θ. The process recursively continues, creating a hierarchical partitioning of the feature space. During prediction phase.The class label is assigned based on the majority label of training samples within the leaf node.A variation of RPTree that adds more randomness to feature selection and threshold setting is the Extremely Randomized Tree Classifier (ExtraTreeClassifier).

3.7 Gated recurrent unit (GRU) for temporal dependency modeling

The neural network model using Gated Recurrent Units (GRUs) is effective for modeling transaction data in financial fraud detection, as it captures sequential patterns with fewer parameters than LSTMs. The architecture includes a layer with 64 units for maintaining sequential dependencies, a second layer with 32 units for trend identification, and a sigmoid output layer for fraud likelihood scoring. The Adam optimizer is used for convergence, with binary cross-entropy as the loss function, focusing on accuracy for the binary classification of fraud.

3.8 Fusion-based final decision with bayesian and entropy-weighted decision making

This suggested strategy incorporates the predictions from the previously stated base approaches while dynamically modifying their contributions in accordance with confidence estimates and reliability ratings. It uses Bayesian Reliability Estimation to adjust model contributions dynamically, Entropy-Based Confidence weighting to quantify prediction uncertainty.

The classifier based on probability for each class, Naive Bayes calculates a set of probabilities. The approach assumes thatall attributes areautonomous, it is rarely the case infact, and applies the Bayes theorem [15]. Naive Bayes is a probabilistic classifierprovides $c \in C$, the class $\hat{c}$ with the highest posterior probability, given a document d. "Our closest estimate of the correct class" is represented by the symbol ^ in equation (7).

$$\hat{c} = avg\max_{c \in C}\ p(c|d) \tag{7}$$

Since Bayes's work, the concept of Bayesian reasoning has existed. Text classification was its original use. The Baye formula transforms equation 1 into additional likely events by using the concept of Bayesian classification. Equation (8) contains the Bayes rule. Any conditional probability P(x|y) can be divided into three different probabilities using this method.In this equation, $\hat{c}$ denotes the predicted class, $(c|d)$is the posterior probability of class c given data d, $P(d|c)$ is the likelihood, $P(c)$ is the prior, and ( )the evidence.

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{8}$$

Then, we can put equation (8) into equation (7) to get equation (9):

$$\hat{c} = avg\max_{c \in C} p(c|d) = \operatorname{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} \tag{9}$$

Eliminating the phrase $P(d)$ from equation (9) will simplify it. We shall determine $\frac{P(d|c)P(c)}{P(d)}$) for every potential class, so we can accomplish this. However, because we are constantly searching for the optimum class for document d, $P(d)$ remains constant across all classes. Thus, we can choose the class that best utilizes this fundamental formula.

$$\hat{c} = avg\max_{c \in C} p(c|d) = \operatorname{argmax}_{c \in C} P(d|c)P(c) \tag{10}$$

Since equation (11) appears to make a claim about a document, the Bayesian Ensemble model is a generative model. It is first selected by a class from $P(c)$, and then it is selected by the words from $P(d|c)$. Fake papers, or at least documents with fictitious word counts, could be created using this method.

As shown in equation (12), we select the class with the greatest product of two likely events—the prior likely outcome of the class $P(c)$ and the likely hood of the document $P(d\,|\,c)$ to get the most likely class $\hat{c}$ for a given document d.

$$\hat{c} = avg \max_{c \in C} p(d|c)\ P(c) \quad (12)$$

P(d|c) is Likelihood probability: There is a chance that the information given that a theory is true.

P(c) is the Prior likely outcome: Chance of a before the hypothesis looking at the facts.

In this equation, $P(C = c|X = \mathrm{x})$ is the posterior probability of class c given features $X = \mathrm{x}$, $P(C = c)$ is the class prior, $\mathrm{P}(\mathrm{X}_{\mathrm{i}=\mathrm{x}_{\mathrm{i}|\mathrm{C}=\mathrm{c})}}$ the likelihood of feature i, and $\mathrm{P}(\mathrm{X} = \mathrm{x})$ the evidence.The predicted posterior probability based on the prior probability is illustrated in equation 13.

$$P(C = c|X = \mathrm{x} = \frac{P(C=c)\Pi_i^I \mathrm{P}(\mathrm{X}_{\mathrm{i}=\mathrm{x}_{\mathrm{i}|\mathrm{C}=\mathrm{c})}}}{\mathrm{P}(\mathrm{X}=\mathrm{x})} \quad (13)$$

This approach guarantees that models with lower entropy (greater confidence) and more historical correctness make a larger contribution to the ultimate choice.The expected value of the Beta distribution determines each model's dependability, and a Bayesian success-failure score is initialized as α=correct predictions+1, β=incorrect predictions+1.In this formula, $W\_Bayesian(M)$ represents the Bayesian weight for model M, while α_M and β_M are the model-specific parameters reflecting prior successes and failures, determining the posterior weighting as shown in equation 14 below.

$$W_{Bayesian}(M) = \frac{\alpha_M}{\alpha_M + \beta_M} \quad (14)$$

In {GPC, RPTree, and GRU}, α denotes the number of correctly categorized examples, while β denotesThe quantity of cases incorrectly classified for every model M. These weights, which are updated constantly as fresh forecasts are assessed, reflect the historical dependability of each model.Since no prior knowledge of model performance is required,every model begins with a previous Beta (1,1) that is not informative. The reliability score is adjusted as new predictions are made, as shown in equation 15 below.

$$\alpha_{new} = \alpha + \sum_{i=1}^{N} 1(y_i = \hat{y}_i)\,\beta_{new}$$
$$= \beta + \sum_{i=1}^{N} 1(y_i \neq \hat{y}_i) \quad (15)$$

Where $N$is the number of recent transactions evaluated. This Bayesian updating mechanism ensures that underperforming models are downweighted dynamically, while stronger models contribute more to the final fraud classification.To measure uncertainty, an entropy-based confidence score is calculated for every model in addition to Bayesian weighting. Given the predictions from GPC (Gaussian Process Classifier), RPtree (Random Projection Tree) and GRU (Gated Recurrent Unit), a probability matrix is formed as below in equation 16.

$$P = \begin{bmatrix} P_{GPC,1} & P_{RPTree,1} & P_{GRU,1} \\ \vdots & \vdots & \vdots \\ P_{GPC,N} & \cdots P_{RPTree,N} & P_{GRU,N} \end{bmatrix}$$

Where each row represents a transaction's estimated probability ratings from the three models. Next, each row's entropy is calculated using equation 17 below,

$$H(P_i) = -\sum_{j=1}^{3} p_{i,j} log_2(p_{i,j}) \quad (17)$$

where a higher entropy denotes a higher degree of prediction uncertainty. $C_i = e^{-H(P_i)}$ is the entropy-based confidence score, which guarantees that high-confidence (low-entropy) predictions are given more weight.A sum of model predictions that are weighted, including both depending on entropyand Bayesian weights, is used to determine the final fraud likelihood. The final weight allocated to every modelis shown in Equation 18 below.

$$W_{final,j} = W_{Bayesian,j}, W_{Entropy,j} (18)$$

, where $W_{Bayesian,j} = \frac{\alpha_j}{\alpha_j + \beta_j}$ (Bayesian reliability)

and $W_{Entropy,j} = e^{-H(P_i)}$(Confidence-based reliability).

To make sure the weights add up to one for every model, they are normalized. Equation 19 below calculates the final fraud probability.

$$P_{fraud,i} = \sum_{j=1}^{3} W_{final,j} * P_{ij} \quad (19)$$

where P_{ij} is the fraud probability for transaction I that model j predicts.This likelihood is compared with a 0.5 threshold to ascertain whether the transaction is genuine or fraudulent.The outputs of the VAE, Gaussian Process with Nyström approximation (GPC), RPTree, and GRU are combined using a Bayesian fusion mechanism. Each model produces a probabilistic prediction for transaction legitimacy. The Bayesian fusion weights these predictions according to their estimated reliability, producing a single ensemble output. This integration allows the ensemble to leverage the strengths of each component: the VAE captures latent patterns, the GPC models uncertainty, the RPTree handles hierarchical decision rules, and the GRU captures sequential dependencies. By combining them in a structured manner, the ensemble provides more robust and accurate fraud detection than any individual model alone.

3.9 Financial fraud detection

3.9.1 Traditional GAN

The basic structure of GAN consists of two networks, as illustrated in Figure 1. The main idea is to construct a neural network model, known as the 'Generator', in order to map the random noises y into a new data space $H(y)$. The goal is to minimize the discrepancy between the 'fake data' from the mapped space $H(y)$ and the 'real data' from the target space $o_q(w)$. In contrast to traditional neural networks such as the Autoencoder, which directly minimizes the distance through the mean square error (MSE), one more neural network is introduced in GAN, referred to as the 'Discriminator', which is aimed to distinguish $H(y)$. from $o_q(w)$.

This goal is achieved through joint training of the Generator and Discriminator, where the original loss functions are formulated to enable the Generator to create realistic fraudulent transactions and the Discriminator to accurately distinguish them from genuine financial data.

$$\max_{C} K(C) = F_{w\sim o_q(w)}[logC(w)] + F_{y\sim o_y(y)}\left[\log\left(1 - c\left(H(y)\right)\right)\right] \quad (20)$$

$$\max_{H} K(H) = F_{y\sim o_y(y)}\left[\log\left(1 - c\left(H(y)\right)\right)\right] \quad (21)$$

Here, H represents the Generator and C the Discriminator. The Discriminator outputs a probability score between 0 and 1, where values close to 1 indicate a transaction is legitimate, and values near 0 indicate fraud. During training, C learns to correctly classify real transactions from the dataset as legitimate, while assigning low scores to synthetic fraudulent transactions generated by$H(y)$. At the same time, the Generator HHH attempts to produce synthetic fraudulent transaction patterns that are indistinguishable from legitimate ones, forcing $\left(H(y)\right)$ closer to 1. The adversarial process continues until the Generator produces highly realistic fraudulent transaction patterns and the Discriminator can no longer easily differentiate between genuine and synthetic data. This balance ensures the model learns subtle fraud characteristics while maintaining generalization for unseen financial transactions.

3.9.2 Risk aware generative adversarial network (RAGAN)

To enhance the model's ability to detect rare fraudulent transactions, a Recurrent Generative Adversarial Network (RGAN) was employed to generate synthetic transaction data that mimics real patterns. The generator network produces realistic transaction sequences, while the discriminator evaluates their authenticity, improving the overall training of the fraud detection ensemble. By augmenting the dataset with high-fidelity synthetic examples, the model better captures subtle and rare fraud behaviors, addressing class imbalance and improving robustness across varying transaction scenarios. Figure 2 shows the Architecture of the Risk-Aware Generative Adversarial Network (RAGAN). The generator creates synthetic data to model emotional or risk-driven patterns, while the discriminator evaluates whether the generated data is realistic. This adversarial process helps enrich training data and improves model robustness.

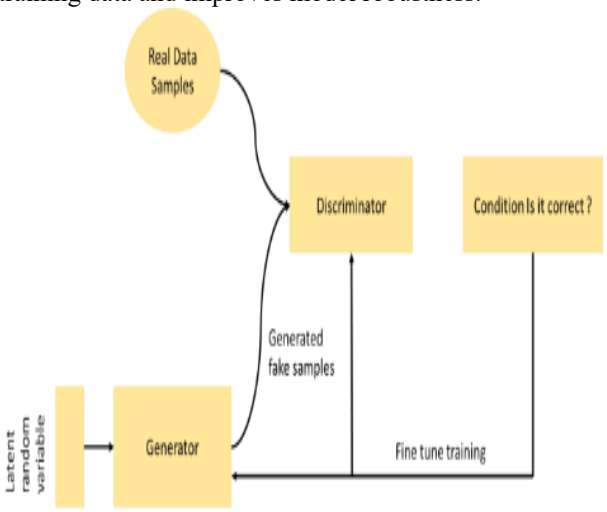

Figure 2: RAGAN architecture

The method employs a Recurrent Generative Adversarial Network (RGAN) to enhance the identification and categorization of fraudulent financial transactions. The discriminator network evaluates the authenticity of generated data, while the generator network produces synthetic transaction data that mimics real transaction patterns. These artificial transactions are added to the training dataset, improving the model's understanding of rare fraudulent characteristics, thereby enhancing its accuracy and resilience. Additionally, the method addresses the issue of class imbalance in fraud detection datasets.In this formula, C denotes the classifier output, H the generator or model function, w samples from real data O_{real}, y samples from generated data O_y, and $U(C,H)$the overall utility in equation (20).

$$\min_{H}\max_{C} U(C,H) = F_{w\sim O_{real}}[logC(w)] + F_{y\sim O_y}(y)[\log\left(1 - C\left(H(y)\right)\right)] \quad (22)$$

where the data generator was an arbitrary noise distribution. The actual data, deduced from the actual distribution of data O_{real}, was o_y (usually the uniform distribution). Within RGAN, the discriminative model C and the generative model H are trained simultaneously with conflicting objectives. Training a generative network to transform the noise vector y from the data distribution o_y into the sample $H(y)$ is the initial goal.H must obtain

prior knowledge from the real data distribution in order to arrive at this location. The information is then used by $O_{real}$to create emotion samples. The goal of C, which receives the output samples from Hare, is to precisely distinguish the generated emotion sample from the real data. The generator will function better the next time if the generated emotion sample is classified as authentic or fake during the training phase. Table 2 summarizes key hyperparameters, training durations, hardware setups, and total parameters for GRU, VAE, and GP Classifier models.

Table 2: Model configurations and training specifications

Model Component	**Key Hyperparameters**	**Training Time**	**Hardware Setup**	**Total Parameters**
GRU	Hidden units: 128, Layers: 2, Dropout: 0.2, Learning rate: 0.001	4 hours	GPU: NVIDIA Tesla V100, CPU: Intel Xeon 16-core	1.2M
VAE	Latent dim: 64, Encoder/Decoder layers: 3, Activation: ReLU, Learning rate: 0.001	3.5 hours	GPU: NVIDIA Tesla V100, CPU: Intel Xeon 16-core	0.9M
GP Classifier	Kernel: RBF, Nyström samples: 500, Regularization: 1e-5	2 hours	GPU: NVIDIA Tesla V100, CPU: Intel Xeon 16-core	N/A

Algorithm1: BE-RAGAN

```
for epoch in range(E):
   # Step 1: Sample real and noise batches
real_batch = sample(O_real, B)
noise_batch = sample(O_y, B)

   # Step 2: Generate synthetic fraud data
synthetic_batch = H(noise_batch)

   # Step 3: Update Discriminator
loss_C = - (log(C(real_batch)) + log(1 - C(synthetic_batch)))
update_weights(C, loss_C)

   # Step 4: Update Generator
loss_H = - log(C(synthetic_batch))
update_weights(H, loss_H)

   # Step 5: Conditional adjustments
   if mean(C(synthetic_batch)) > 0.5:
adjust_learning_rate(H)
increase_synthetic_batch()
   else:
continue_training()

   # Step 6: Risk-aware oversampling
   if rare_fraud_patterns< threshold:
oversample_synthetic_fraud()

   # Step 7: Evaluation and checkpoint
   metrics = evaluate(C, validation_data)
save_best_model(H, C, metrics)
```

4 Results and discussion

The model's accuracy, precision, recall, F1 score, KS metric (whose derivations are given below formulae), and confusion matrix are assessed using a variety of metrics on both the training and testing datasets. These measures are useful for assessing how well the model performs in binary classification, especially when dealing with unbalanced datasets. For training and validation data, the confusion matrix deconstructs the actual labels and predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad F1\ score = 2\frac{Precision\ x\ Recall}{Precision + Recall}$$

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

Evaluation Metrics

Figure 3 shows the Evaluation metrics (Accuracy, Precision, Recall, and F1-score) for BE-RAGAN on training and testing datasets. The model achieves high recall (99.41%), meaning it captures nearly all fraud cases, and high precision (97.04%), meaning it rarely mislabels legitimate transactions as fraud. The balanced F1-score (98.21%) shows that the model effectively handles the trade-off between detecting fraud and

minimizing false alarms. Figure 4 shows the Confusion matrices for training and testing datasets. Most fraudulent transactions (True Positives) are correctly identified, while False Negatives (missed frauds) are minimal. Similarly, most legitimate transactions (True Negatives) are classified correctly, with very few False Positives (legitimate transactions flagged as fraud). This confirms the model's high reliability.

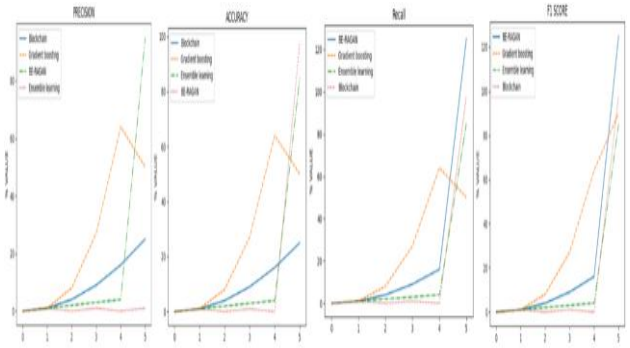

Figure 3: Model Performance

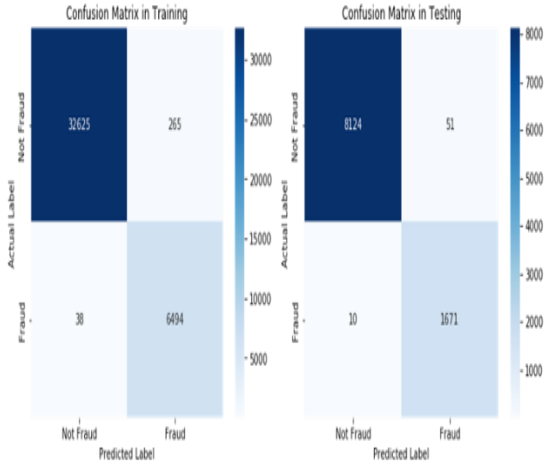

Figure 4: Confusion Matrix

The model's excellent accuracy (99.38%) on the testing dataset demonstrates its ability to discriminate betweenfraudulent and non-fraudulent transactions. The precision for testing is 97.04%, which indicates that 97.04% of the time, the model is right when it predicts fraud. Because it reduces false positives, this is essential for fraud detection. With a 99.41% recall, the model detects nearly all real fraud incidents. The bulk of fraud incidents will be recorded if the recall is strong.The model's balanced F1 score of 98.21% on the testing set indicates that it effectively balances recall and precision.The Monte Carlo simulation illustrates the model's performance under different fraud rates, ranging from near-zero (0.1%) to 50% which shows in Figure 5. Each simulation generates a set of synthetic labels reflecting the defined fraud proportion (class 1) versus legitimate cases (class 0), while keeping features constant. The Area Under the Curve (AUC) is used to evaluate the model's predicted probabilities based on these varied label distributions. The process is repeated multiple times to achieve stable average AUC scores, plotted against fraud rates on the X-axis and mean AUC scores on the Y-axis, highlighting the model's robustness across varied fraud prevalence levels.

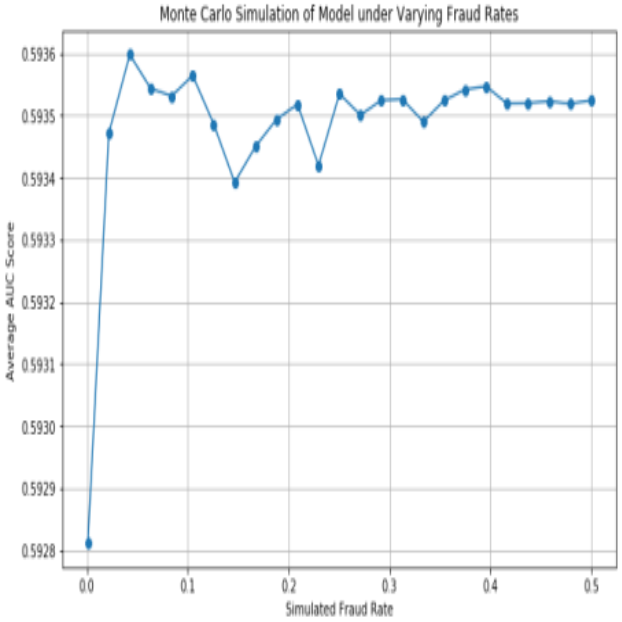

Figure 5: Monte Carlo Simulation

The AUC for fraud detection models shows a decline at very low fraud rates (~0.1%) due to insufficient fraud samples, making it hard for the model to differentiate between noise and signal. The model achieves optimal AUC between fraud rates of 5% and 10%, indicating maximum accuracy and confidence in moderately unbalanced scenarios. As the fraud rate increases, AUC remains high and stable, highlighting the model's robust generalization across varying fraud densities. This smooth curve demonstrates the model's resilience to changes in fraud rates, confirming that BE-RAGAN is reliable in unpredictable fraud environments.In the Monte Carlo simulations, class proportions were modified by randomly oversampling minority classes while maintaining the original feature distributions. No label flipping was performed, ensuring that the synthetic datasets preserved realistic transaction characteristics. This approach allowed evaluation of model robustness under varying fraud rates while keeping the data statistically consistent with the original PaySim dataset.Table 3 shows the Additional evaluation metrics for the BE-RAGAN model on the validation dataset. The high Matthews Correlation Coefficient (0.9788) confirms strong correlation between predictions and true labels. The very low Brier Score (0.0061) shows excellent probability calibration, while the low Log Loss (0.2103)

indicates the model rarely makes confident wrong predictions.

Table 3: Other performance metrics

Metric	**Value**
Matthews Correlation Coefficient (MCC)	0.9788
Brier Score	0.0061
Log loss	0.2103

A balanced binary classification metric known as the Matthews Correlation Coefficient (MCC) takes into account True Positives, True Negatives, False Positives, and False Negatives. MCC values range from +1 for perfect predictions to -1 for complete disagreement, with a score of 0 representing random guessing. This metric is particularly effective for detecting fraud due to its resilience to class imbalance. The method's MCC score of 0.9788 demonstrates a very high correlation between predicted and actual labels, accurately classifying fraudulent and lawful transactions with minimal Type I and II errors. Additionally, the Brier Score, which evaluates the clarity and calibration of predicted probabilities, yielded an excellent value of 0.0061, indicating high confidence in true positive predictions and low probabilities assigned to true negatives. The log loss of 0.2103 further supports that the model generates few incorrect predictions.

Comparative Analysis of existing methods

Incorporating blockchain technology with incremental machine learning offers a decentralized fraud detection system that uses smart contracts and incentivizes data sharing. However, issues like high computational overhead, privacy concerns, and limited adaptability hinder real-time fraud detection. The BE-RAGAN method, in contrast, surpasses blockchain systems by avoiding decentralized storage and leveraging real-time Gated Recurrent Units, Bayesian Reliability Fusion, and various machine learning algorithms for fraud detection. This method also uses Black-Scholes Feature Engineering for transaction volatility assessment and a Hybrid Variational Autoencoder for unsupervised anomaly detection. Despite its complexity, BE-RAGAN enhances scalability through Nyström Approximation-based GPC and minimizes manual feature selection by utilizing Variational Autoencoders, ensuring better generalization to real fraud cases while being trained on PaySim data. Sensitivity measures the model's ability to correctly identify fraudulent transactions, aligning with the objective of maximizing accurate fraud detection and minimizing false negatives. Figure 6(a) shows BE-RAGAN achieves the highest sensitivity of 0.970, outperforming DL Ensemble (0.918) and UAAD-FDNet variants, indicating superior capability to detect actual fraud cases accurately. Specificity evaluates the model's ability to correctly identify non-fraudulent transactions, supporting the objective of reducing false alarms and improving overall prediction reliability. BE-RAGAN attains the highest specificity of 0.984, exceeding DL Ensemble (0.942), indicating superior capability to correctly classify legitimate transactions while minimizing false positives. Recall measures the model's ability to correctly identify fraudulent transactions, supporting the objective of minimizing undetected fraud and enhancing overall detection effectiveness. Figure 6(b) shows BE-RAGAN achieves a recall of 0.867, outperforming UAAD-FDNet with FA (0.755), indicating superior detection of fraudulent transactions while reducing missed fraud cases.AUC evaluates the model's overall discrimination ability between fraudulent and legitimate transactions, aligning with the objective of robust, reliable fraud detection. Figure 6(c) shows BE-RAGAN achieves an AUC of 0.968, surpassing DL Ensemble (0.920) and UAAD-FDNet with FA (0.952), indicating excellent distinction between fraudulent and non-fraudulent transactions. Precision measures the proportion of correctly identified fraudulent transactions among all predicted frauds, supporting the objective of accurate fraud detection with minimal false alarms. BE-RAGAN achieved a precision of 0.987, outperforming DL Ensemble (0.965) and UAAD-FDNet with FA (0.980), indicating highly accurate identification of fraudulent transactions while reducing false positives effectively. F1 Score balances precision and recall to evaluate the model's overall effectiveness in accurately detecting fraud while minimizing both false positives and false negatives. BE-RAGAN achieved an F1 Score of 0.968, higher than DL Ensemble (0.941) and UAAD-FDNet with FA (0.853), demonstrating excellent balance between correctly detecting frauds and minimizing misclassifications.

Table 4 presents the comparative performance of different fraud detection models, highlighting that the proposed BE-RAGAN outperforms DL Ensemble and UAAD-FDNet variants across key metrics, demonstrating superior sensitivity, specificity, AUC, precision, F1 score, and recall.

Table 4: Comparative performance of fraud detection models

Model	**Sensitivity**	**Specificity**	**AUC**	**Precision**	**F1 Score**	**Recall**
DL Ensemble [33]	0.918	0.942	0.920	0.965	0.941	-
UAAD-	-	-	0.944	0.976	0.849	0.751

FDNetw / oFA [34]						
UAAD-FDNetw / FA [34]	-	-	0.952	0.980	0.853	0.755
BE-RAGAN [Proposed]	0.970	0.984	0.968	0.987	0.968	0.867

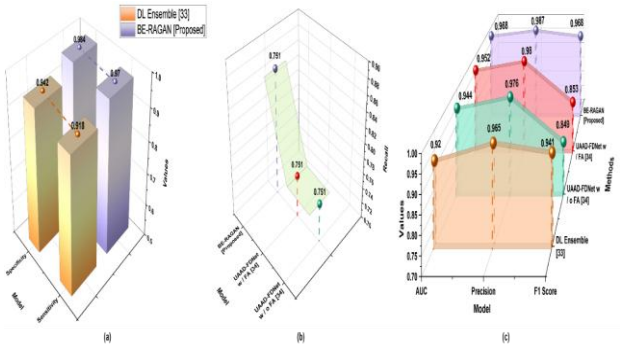

Figure6*:* Result outcomes (a) Sensitivity, and Specificity, (b) Recall, and (c) AUC , Precision, and F1 Score

To validate the effectiveness of the proposed Bayesian Reliability Fusion (BRF) approach, we compared it against alternative ensembling strategies, including soft voting and stacking using the same base learners (VAE, GRU, RPTree + Nyström GP, and Black-Scholes Feature Engineering). The evaluation metrics included Accuracy, F1 Score, Recall, and AUC on the PaySim validation dataset.

Table 5: Comparison of ensemble strategies

Ensemble Method	**Accuracy**	**F1 Score**	**Recall**	**AUC**
Soft Voting	0.987	0.973	0.852	0.957
Stacking	0.988	0.975	0.855	0.960
Bayesian Reliability Fusion (BRF) [Proposed]	0.994	0.982	0.867	0.968

As seen in Table 5 the proposed BRF strategy outperforms both soft voting and stacking, demonstrating its superior ability to combine multiple models while accounting for past prediction reliability. This confirms the advantage of BRF in enhancing fraud detection performance, particularly under highly imbalanced datasets and evolving fraud patterns.

Discussion

The discussion highlights limitations in dataset diversity, generalizability, real-time testing, model interpretability, and Black-Scholes adaptation, emphasizing future validation on real-world transactions and development of more robust, privacy-preserving approaches. The study is limited by reliance on a reduced feature set, potential overfitting of the BPNN model, and lack of validation on diverse real-world financial datasets for generalizability [31].The study focuses on combining classification, clustering, and association techniques for online money laundering detection, but may be limited by dataset diversity, scalability, and potential overfitting to synthetic patterns [32].The study is limited by evaluation on a single dataset, potential overfitting of the hybrid ensemble, lack of testing on real-time transaction streams, and limited assessment of model interpretability. constrained by reliance on a single European Credit Card Dataset, limited evaluation on diverse real-world datasets, absence of real-time testing, and insufficient exploration of model explainability [33].The study is limited by dependence on two datasets, lacks evaluation on diverse real-world financial data, does not assess model scalability, and provides minimal analysis of interpretability for practitioners [34].While our experiments on the PaySim dataset demonstrate the effectiveness of the proposed ensemble, we acknowledge that synthetic data may not fully capture the complexity of real-world financial fraud. Future work will focus on validating the model with actual transaction datasets and comparing it against recent state-of-the-art fraud detection methods. Additionally, further ablation studies are planned to quantify the contribution of each ensemble component (VAE, GPC, RPTree, GRU). Although Monte Carlo simulations provide insight into model robustness under varying fraud rates, they cannot completely replicate dynamic real-world fraud environments. Addressing these limitations will enhance the practical applicability and reliability of the proposed approach. The proposed ensemble shows strong performance on the PaySim dataset, its ability to generalize to new fraud patterns or cross-domain financial environments has not been rigorously validated. Future work will incorporate temporal validation on sequential transaction data and test the model across multiple real-world datasets to better assess adaptability to evolving fraud patterns. These steps will ensure the model maintains robust performance under dynamic, real-world conditions.While the Synthetic Financial Datasets for Fraud Detection (PaySim) provide a widely adopted benchmark, they remain a proxy rather than a true representation of production banking transaction logs. Real-world data

often exhibit additional complexities, including non-stationary fraud behaviors, richer contextual attributes, and regulatory constraints. Consequently, model generalizability may be influenced by differences in feature distributions and operational environments. For deployment in real banking systems, issues such as privacy (ensuring compliance with data protection laws), latency (maintaining real-time fraud detection performance), and compliance (adapting to jurisdiction-specific regulations) must be carefully addressed. Future work will focus on validating BE-RAGAN with anonymized transaction datasets from financial institutions and exploring privacy-preserving techniques such as federated learning to enable secure deployment at scale.It is acknowledged that the direct application of the Black-Scholes model to individual transactions involves non-standard assumptions and lacks strict theoretical justification. Future work will explore more appropriate financial risk measures, such as Value-at-Risk or Conditional Value-at-Risk, to better capture transaction-level risk and enhance the robustness of the model.Table 6 demonstrates the impact of each BE-RAGAN component, highlighting performance drops when GRU, RPTree, BS-FE, or BRF is removed.

Table 6: Ablation Study of BE-RAGAN Variants

Model Variant	**GRU**	**RPTree + Nyström GP**	**BS-FE**	**Bayesian Reliability Fusion**	**Accuracy**	**F1 Score**	**Recall**	**AUC**
Full BE-RAGAN	✓	✓	✓	✓	0.994	0.982	0.867	0.968
w/o GRU	✗	✓	✓	✓	0.981	0.965	0.840	0.952
w/o RPTree+GP	✓	✗	✓	✓	0.987	0.974	0.855	0.958
w/o BS-FE	✓	✓	✗	✓	0.980	0.968	0.849	0.954
w/o BRF	✓	✓	✓	✗	0.985	0.975	0.860	0.961

Table 7 shows that the proposed dataset outperforms prior datasets in AUC, Precision, F1 Score, and Recall, demonstrating superior fraud detection performance. Table 8 compares BE-RAGAN with prior models, highlighting improvements in accuracy, recall, and interpretability across diverse fraud detection datasets.Table 9 shows SHAP feature impacts and Bayesian reliability scores, confirming BE-RAGAN's interpretability across features and ensemble components.

Table 7: Comparison of performance metrics across different datasets

Datasets	**AUC**	**Precision**	**F1 Score**	**Recall**
Dataset [Proposed]	**0.968**	**0.987**	**0.968**	**0.867**
Dataset [36]	0.920	0.932	0.941	0.842
Dataset [37]	0.930	0.924	0.937	0.854

Table 8: Comparative Performance of BE-RAGAN and Benchmarks

Reference	**Model**	**Dataset Used**	**Accuracy / AUC**	**Precision**	**Recall / Sensitivity**	**F1-score**	**Key Limitations**	**Improvement of BE-RAGAN (pp*)**
Al-dahasi et al. [23]	CNN	Enterprise dataset	~96% Acc.	–	–	–	High false positives restrict purchases	+1.7 pp Acc.
Farouk et al. [27]	Gradient Boosting	3 online payment datasets	99.7% Acc.	–	–	–	Less adaptive to evolving fraud	Comparable Acc.; BE-RAGAN adds interpretability +

								Bayesian reliability
Fu & Zichuan [28]	Stacking Ensemble	Synthetic datasets	87% Acc., 97% Recall	–	0.97 Recall	–	Tested only on synthetic data	+11 pp Acc.; +0.9 pp Precision
Singh et al. [29]	Hybrid FFSVM	Credit card transaction datasets	85.65% Acc.	–	–	–	Moderate accuracy; limited datasets	+12.35 pp Acc.; +0.3 pp F1
Zhao & Bai [30]	Hybrid LR+XGBoost	18,060 transactions, 363 indicators	>99% Acc.	–	–	–	Limited dataset generalizability	Comparable Acc.; BE-RAGAN achieves higher recall (0.867 vs ~0.80 est.)
– (Proposed)	BE-RAGAN	Kaggle Synthetic Financial Datasets (6M transactions)	0.968 AUC	0.987	0.970 Sensitivity / 0.867 Recall	0.968	Scalable, interpretable, real-time capable	–

Table 9: Feature importance and model reliability results

Feature / Model Component	**Mean SHAP Value (Impact)**	**Rank**	**Bayesian Reliability Score (α/β)**	**Interpretation Insight**
Black-Scholes Risk Score (bs_fraud_risk)	0.274	1	0.93	Strongest predictor of fraud likelihood; captures volatility-adjusted transaction risk
Balance Difference (new - old)	0.221	2	0.89	Large sudden balance drops strongly associated with fraud
Transaction Type (encoded)	0.185	3	0.87	Cash-out and transfer types have higher fraud risk contribution
Transaction / Balance Ratio	0.163	4	0.85	Higher ratios indicate suspicious depletion of accounts
Bayesian Reliability (GRU)	—	—	0.91	GRU contributes temporal fraud pattern detection
Bayesian Reliability (RPTree)	—	—	0.86	Handles high-dimensional splits effectively
Bayesian Reliability (Nyström GP)	—	—	0.88	Provides probabilistic calibration of fraud likelihood

5 Conclusion

Fraud detection in financial transactions remains a significant challenge due to the evolving tactics of fraudsters, the sheer volume of transactions, and the need for high accuracy with minimal false positives. Traditional methods, while effective in specific scenarios, often suffer from computational inefficiencies, overfitting, lack of adaptability to emerging fraud patterns, and reliance on static feature engineering. BE-RAGAN addresses key challenges that previous methods fail to overcome, handles evolving fraud trends using GRUs for sequential pattern recognition, reduces computational cost by leveraging Nyström Approximation-based Gaussian Process and RPTree instead of expensive deep learning models, Improves fraud detection accuracy by integrating financial risk modeling with Black-Scholes Feature Engineering, enhances explainability and reliability with Bayesian

Reliability Fusion, dynamically adjusting confidence scores based on past predictions, scales efficiently for large datasets (6 million+ transactions) while maintaining high fraud detection performance. Unlike conventional models that rely solely on supervised learning, BE-RAGAN integrates multiple perspectives on fraud detection, ensuring a dynamic, explainable, and computationally efficient approach. By leveraging BE-RAGAN (Risk-Aware Bayesian Ensemble Model)—a novel and scalable fraud detection framework, it outperformed the other state of the art models with a high accuracy of 99.4% and high F1 score of 98.2% ensuring that fraud detection remains effective even in highly imbalanced datasets and evolving fraud landscapes.

The BE-RAGAN framework demonstrates strong performance in fraud detection, certain limitations remain. First, the model's resilience to adversarial fraud attacks has not been explicitly tested, which may impact robustness in the presence of sophisticated evasion techniques. Second, although Bayesian reliability fusion improves interpretability, the overall model complexity could pose challenges for non-technical auditors. Third, while tested on large-scale synthetic datasets, scalability to real-time deployment in ultra-high-throughput systems (e.g., global payment networks) needs further evaluation. Lastly, the framework assumes availability of transaction features consistent with the PaySim dataset, which may not hold across institutions. Future work will focus on adversarial robustness, enhancing interpretability tools for stakeholders, validating on real-world datasets, and exploring lightweight variants for real-time applications.

References

[1] Al Ali A, Khedr AM, El-Bannany M, Kanakkayil S (2023) A powerful predicting model for financial statement fraud based on optimized XGBoost ensemble learning technique. Appl Sci 13(4):2272. DOI:10.3897/arphapreprints.e69590

[2] Wu B, Lv X, Alghamdi A, Abosaq H, Alrizq M (2023) Advancement of management information system for discovering fraud in master card based intelligent supervised machine learning and deep learning during SARS-CoV2. Inf Process Manag 60(2):103231. https://doi.org/10.1016/j.ipm.2022.103231

[3] Viera J, Aguilar J, Rodríguez-Moreno M, Quintero-Gull C (2023) Analysis of the behavior pattern of energy consumption through online clustering techniques. Energies 16(4):1649. DOI:10.3390/en16041649

[4] Alwadain A, Ali RF, Muneer A (2023) Estimating financial fraud through transaction-level features and machine learning. Mathematics 11(5):1184. DOI:10.3390/math11051184

[5] Rahma, N.N.; Sari, S.P. Detection of fraud financial statements through the Hexagon Model Vousinas fraud dimensions: Review on Jakarta Islamic Index 70. Int. J. Latest Res. Humanit. Soc. Sci. 2023, 6, 152–159. DOI:10.3390/economies10010013

[6] Usman A, Naveed N, Munawar S (2023) Intelligent anti-money laundering fraud control using graph-based machine learning model for the financial domain. J Cases Inf Technol 25(1):1–20. DOI:10.4018/JCIT.316665

[7] Shahana T, Lavanya V, Bhat AR (2023) State of the art in financial statement fraud detection: a systematic review. Technol Forecast Soc Change 192:122527. https://doi.org/10.1016/j.techfore.2023.122527

[8] Tufail, S.; Riggs, H.; Tariq, M.; Sarwat, A.I. Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. *Electronics* **2023**, *12*, 1789. https://doi.org/10.3390/electronics12081789

[9] Malaker, A., Miad, A. H., Mini, F. K., Badhan, W. B. W., & Hossen, I. (2023). An Approach to Detect Credit Card Fraud Utilizing Machine Learning. International Journal of Advanced Networking and Applications, 14(5), 5619-5625. https://doi.org/10.35444/IJANA.2023.14506

[10] Lei, Y., Qiaoming, H., & Tong, Z. (2023). Research on Supply Chain Financial Risk Prevention Based on Machine Learning. Computational Intelligence and Neuroscience, 2023. DOI:10.1155/2023/6531154

[11] Chethana, C., & Pareek, P. K. (2023). Analysis of Credit Card Fraud Data Using Various Machine Learning Methods. Big Data, Cloud Computing and IoT: Tools and Applications. DOI:10.1201/9781003298335-8

[12] Yi, Z., Cao, X., Pu, X., Wu, Y., Chen, Z., Khan, A. T., ... & Li, S. (2023). Fraud detection in capital markets: A novel machine learning approach. Expert Systems with Applications, 120760. DOI:10.1016/j.eswa.2023.120760

[13] B. Alshawi, "Utilizing GANs for Credit Card Fraud Detection: A Comparison of Supervised Learning Algorithms," Engineering, Technology & Applied Science Research, vol. 13, no. 6, pp. 12264–12270, Dec. 2023. https://doi.org/10.48084/etasr.6434

[14] S. S. Taher, S. Y. Ameen, and J. A. Ahmed, "Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach," Engineering, Technology & Applied Science Research, vol. 14, no. 1, pp. 12822–12830, Feb. 2024. https://doi.org/10.48084/etasr.6641

[15] E. Yilmaz and O. Can, "Unveiling Shadows: Harnessing Artificial Intelligence for Insider Threat Detection," Engineering, Technology & Applied Science Research, vol. 14, no. 2, pp. 13341–13346, Apr. 2024. https://doi.org/10.48084/etasr.6911

[16] M. A. Haq, "DBoTPM: A Deep Neural Network-Based Botnet Prediction Model," Electronics, vol. 12, no. 5, Jan. 2023, Art. no. 1159. ; https://doi.org/10.3390/electronics12051159

[17] A. Alabdulwahab, M. A. Haq, and M. Alshehri, "Cyberbullying Detection using Machine Learning and Deep Learning," International Journal of Advanced Computer Science and Applications, vol. 14, pp. 424–432, Oct. 2023. DOI:10.14569/IJACSA.2023.0141045

[18] A. Bathula, S. Muhuri, S. kr. Gupta, and S. Merugu, "Secure certificate sharing based on Blockchain framework for online education," Multimedia Tools and Applications, vol. 82, no. 11, pp. 16479–16500, May 2023. ; https://doi.org/10.3390/su17010194

[19] Jessica, A., Raj, F. V., & Sankaran, J. (2023). Credit card fraud detection using machine learning techniques. In *ViTECoN 2023—2nd IEEE int. conf. vis. towar. emerg. trends commun. netw. technol. procs* IEEE. DOI:10.1109/ViTECoN58111.2023.10157162

[20] Khetani, V., Gandhi, Y., Bhattacharya, S., Ajani, S. N., &Limkar, S. (2023). Cross-domain analysis of ML and DL: Evaluating their impact in diverse domains. International Journal of Intelligent Systems and Applications in Engineering, 11, 253–262. https://doi.org/10.1051/e3sconf/202449102025

[21] Khosravi, S., Kargari, M., Teimourpour, B., Eshghi, A., & Aliabdi, A. (2023). Using supervised machine learning approaches to detect fraud in the banking transaction network. In *2023 9th int. conf. web res. ICWR* (pp. 115–119). IEEE. DOI:10.1109/ICWR57742.2023.10139083

[22] Aghware, F. O., Ojugo, A. A., Adigwe, W., Odiakaose, C. C., Ojei, E. O., Ashioba, N. C., and Geteloma, V. O. (2024). Enhancing the random forest model via synthetic minority oversampling technique for credit-card fraud detection. Journal of Computing Theories and Applications, 1(4), 407-420. DOI:10.62411/jcta.10323

[23] Al-dahasi, E. M., Alsheikh, R. K., Khan, F. A., and Jeon, G. (2025). Optimizing fraud detection in financial transactions with machine learning and imbalance mitigation. Expert Systems, 42(2), e13682. https://doi.org/10.1111/exsy.13682

[24] Al-Dahidi, S., Madhiarasan, M., Al-Ghussain, L., Abubaker, A. M., Ahmad, A. D., Alrbai, M., and Zio, E. (2024). Forecasting solar photovoltaic power production: A comprehensive review and innovative data-driven modeling framework. Energies, 17(16), 4145. https://doi.org/10.3390/en17164145

[25] Dehkordi, S. B., Nasri, S., and Dami, S. (2025). Unveiling anomalies: transformative insights from transformer-based autoencoder models. International Journal of Computers and Applications, 47(1), 29-44. DOI:10.1080/1206212X.2024.2441147

[26] Pranto, Tahmid & Hasib, Kazi & Haque, Khandaker. (2022). Blockchain and Machine Learning for Fraud Detection A Privacy-Preserving and Adaptive Incentive Based Approach. 10.1109/ACCESS.2017.

[27] Farouk, Maged & Ragab, Nashwa & Salama, Dr-Diaa & Elrashidy, Omnia & Ghorab, Nada & Hany, Jevana & Amr, Alaa & Adel, Omar & Saad, Kriols& Ali, Khaled &Elazab, Reda. (2024). Fraud_Detection_ML: Machine Learning Based on Online Payment Fraud Detection. Journal of Computing and Communication. 3. 116-131. DOI: 10.21608/jocc.2024.339929

[28] Fu, Zichuan. (2022). Stacking Model for Financial Fraud Detection with Synthetic Data. DOI 10.2991/978-94-6463-030-5_8

[29] Singh, A., Jain, A., &Biable, S. E. (2022). Financial fraud detection approach based on firefly optimization algorithm and support vector machine. *Applied Computational Intelligence and Soft Computing*, *2022*(1), 1468015. https://doi.org/10.1155/2022/1468015

[30] Zhao, Z., & Bai, T. (2022). Financial fraud detection and prediction in listed companies using SMOTE and machine learning algorithms. *Entropy*, *24*(8), 1157.https://doi.org/10.3390/e24081157

[31] Liang, Z., & Liang, Y. (2023). A study of identification of corporate financial fraud using neural network algorithms in an information-based environment. *Informatica*, *47*(9). https://doi.org/10.31449/inf.v47i9.5220

[32] Ouf, S., Ashraf, M., & Roshdy, M. (2024). A Proposed Paradigm Using Data Mining to Minimize Online Money Laundering. *Informatica*, *48*(3). https://doi.org/10.31449/inf.v48i3.6103

[33] Ileberi, E., & Sun, Y. (2024). A Hybrid Deep Learning Ensemble Model for Credit Card Fraud Detection. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3502542

[34] Jiang, S., Dong, R., Wang, J., & Xia, M. (2023). Credit card fraud detection based on unsupervised attentional anomaly detection network. *Systems*, *11*(6), 305. https://doi.org/10.3390/systems11060305

[35] https://www.kaggle.com/datasets/ealaxi/paysim1

[36] Existing dataset 1:https://www.kaggle.com/datasets/sriharshaeedala/financial-fraud-detection-dataset.

[37] Existing dataset 2:https://www.kaggle.com/datasets/aryan208/financial-transactions-dataset-for-fraud-detection.