

Article

Advanced Tax Fraud Detection: A Soft-Voting Ensemble Based on GAN and Encoder Architecture

Masad A. Alrasheedi ¹, Samia Ijaz ^{2,*}, Ayed M. Alrashdi ³ and Seung-Won Lee ^{4,5,6,7,*}

¹ Department of Management Information Systems, College of Business Administration, Taibah University, Al-Madinah Al-Munawara 42353, Saudi Arabia; mrshedi@taibahu.edu.sa

² Department of Computer Science, HITEC University, Taxila 47080, Pakistan

³ Department of Electrical Engineering, College of Engineering, University of Ha'il, Ha'il 81441, Saudi Arabia; am.alrashdi@uoh.edu.sa

⁴ Department of Precision Medicine, Sungkyunkwan University School of Medicine, Suwon 16419, Republic of Korea

⁵ Department of Metabiohealth, Sungkyunkwan University, Suwon 16419, Republic of Korea

⁶ Personalized Cancer Immunotherapy Research Center, Sungkyunkwan University School of Medicine, Suwon 16419, Republic of Korea

⁷ Department of Artificial Intelligence, Sungkyunkwan University, Suwon 16419, Republic of Korea

* Correspondence: samia.ijaz@hitecuni.edu.pk (S.I.); swleemd@g.skku.edu (S.-W.L.)

Abstract: The world prevalence of the two types of authorized and fraudulent transactions makes it difficult to distinguish between the two operations. The small percentage of fraudulent transactions, in turn, gives rise to the class imbalance problem. Hence, an adequately robust fraud detection mechanism must exist for tax systems to avoid their collapse. It has become significantly difficult to obtain any dataset, specifically a tax return dataset, because of the rising importance of privacy in a society where people generally feel squeamish about sharing personal information. Because of this, we arrive at the decision to synthesize our dataset by employing publicly available data, as well as enhance them through Correlational Generative Adversarial Networks (CGANs) and the Synthetic Minority Oversampling Technique (SMOTE). The proposed method includes a preprocessing stage to denoise the data and identify anomalies, outliers, and dimensionality reduction. Then the data have undergone enhancement using the SMOTE and the proposed CGAN techniques. A unique encoder design has been proposed, which serves the purpose of exposing the hidden patterns among legitimate and fraudulent records. This research found anomalous deductions, income inconsistencies, recurrent transaction manipulations, and irregular filing practices that distinguish fraudulent from valid tax records. These patterns are identified by encoder-based feature extraction and synthetic data augmentation. Several machine learning classifiers, along with a voting ensemble technique, have been used both with and without data augmentation. Experimental results have shown that the proposed Soft-Voting technique outperformed the original without an ensemble method.

Keywords: soft-voting ensemble; tax evasion; synthetic dataset; fraud detection; SMOTE; GAN; financial statement

MSC: 68T07



Academic Editor: Raymond Lee

Received: 15 January 2025

Revised: 13 February 2025

Accepted: 14 February 2025

Published: 16 February 2025

Citation: Alrasheedi, M.A.; Ijaz, S.; Alrashdi, A.M.; Lee, S.-W. Advanced Tax Fraud Detection: A Soft-Voting Ensemble Based on GAN and Encoder Architecture. *Mathematics* **2025**, *13*, 642. <https://doi.org/10.3390/math13040642>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Tax fraud costs governments and financial organizations globally a lot of money. Traditional rule-based auditing systems cannot keep up with sophisticated fraud, requiring powerful machine learning techniques. The class imbalance problem, where fraudulent

transactions make up only a small fraction of tax records, makes it difficult for machine learning models to learn meaningful fraud patterns, and the lack of publicly available tax fraud datasets, as tax return data are highly sensitive and legally protected, limits real-world data for training fraud detection models. Current fraud detection systems fail to identify fraudsters and misidentify honest taxpayers without solving these difficulties. Every business unit has adopted data as a core asset, including the state. Modern technologies have allowed for the creation of a range of different data products that proactively target the improvement of a company's competitive advantage. With all the perks that data bring, the clamoring for its use is on the rise, so regulating data use must be a part of privacy rights. In the name of privacy, different laws and regulations have been rolled out.

Suppose an individual or a corporate body finds a way of avoiding paying taxes by underreporting income or overstating deductions and exemptions. This act is called "tax evasion", and it is considered an illegal act [1–3]. In addition, tax evasion is sometimes addressed as "shadow economy" because of the revenue loss to which governments fall prey when they construct their economic and social policy. Moreover, they are bound to allocate this revenue for the purpose of fraud detection [4–7]. Furthermore, it is irrational for this practice not to hurt honest taxpayers [8]. Hence, it is an unavoidable obligation of the state to combat deception. As per statistics, in 2021, multinational corporations and individuals abused taxes that led the world economy to lose more than 480 billion USD. That loss potentially could vaccinate the entire world's population by three types. Given the amount of tax declarations and documentation received from organizations, an audit requires the employment of technology.

Whereas tax evasion schemes change over time [9,10], the integrity of tax officials is not as consistent, and bribery is in our present world. Therefore, tax auditing duties can be complemented, and tax evasion could be discouraged due to the busy work that would require digitalization [11]. Business Intelligence (BI) assists tax personnel in screening financial statements and distinguishing those worthy of further audit. In addition, the use of machine learning would help the officials resolve these problems. In this case, the model reveals insights that are much more profound than the formal risk criteria could. The categories of machine learning are mainly two: supervised and unsupervised. The main difference between the two lies in providing labels while training the model. That is, supervised learning requires labeled data while setting up a model, whereas in unsupervised learning, the labeled data are unnecessary [12,13]. The digitalization and incorporation of machine learning can help in enhancing the communication of tax information by officials from one working department or area to another, i.e., investigating fresh cases of tax and tax evasion arising over time [9]. There is yet another classification into further machine learning techniques like reinforcement learning and semi-supervised learning [12,14,15]. These methods may also well match the applications mentioned.

The increasing number of laws and regulations tightly controls financial data usage. Nevertheless, there are multiple ways in which such data may be obtained in the public domain. There is public access to datasets, which may be obtained from platforms such as Kaggle, the UCI Repository, and Google Data Search, or it may be available from some other capacitated organizations that operate using a consensus mechanism whereby such data are open to everyone to access. Annual reports of the organizations provide open-access data and can be available to investors and stakeholders as well. It is important that attention is given to the assessment of open-source data for quality and reliability to ensure reliable results. This will involve checking and rating a few conditions, such as what data type and source are mentioned, when it was released, and the contact information in case a need arises for any further inquiries. In our study, we review the financial statements of Stock Exchange of Thailand (SET)-listed companies. In these markets, it is also common

for that financial data to come from companies listed on international stock exchanges. Such datasets are adaptable for specific research questions and thereby present a great deal of potential for personalized analysis. Researchers will be able to access a wide range of financial data subject to legal and regulatory constraints through such resources.

Global tax evasion costs billions of dollars and strains government finances and public services [16]. Rule-based auditing and manual inspections cannot keep up with tax evaders' increasingly sophisticated strategies [17,18]. Machine learning is a great fraud detection technology, but it faces two key obstacles. First, the severe class imbalance problem—where fraudulent transactions make up a small percentage of records—causes models to prefer legal transactions, resulting in poor fraud detection [19]. Second, tax return data are sensitive and protected by privacy laws, making real tax fraud datasets difficult to get for research and model development [20]. Without addressing these issues, present systems would miss fraudulent cases, causing financial losses and tax enforcement agency distrust.

This study addresses class imbalance in fraudulent transactions and the lack of real tax fraud datasets to develop a revolutionary tax fraud detection method. This research's main contributions:

- Detecting tax fraud with synthetic data: we develop high-quality synthetic fraudulent data using Correlational Generative Adversarial Networks (CGANs) and Synthetic Minority Over-sampling Technique (SMOTE) to overcome data scarcity and improve model resilience.
- A new encoder architecture exposes hidden patterns in tax transaction records, improving fraud detection accuracy.
- Soft-Voting Ensemble Model: a weighted majority voting ensemble with numerous machine learning classifiers improves fraud categorization and reduces misclassification.
- Comprehensive performance evaluation: the proposed system outperforms standard classifiers in precision, recall, F1 score, and AUROC across multiple datasets.

Using synthetic data and ensemble learning to detect tax fraud is the key contribution of this research. The Soft-Voting Ensemble approach and synthetic dataset production method (CGAN and SMOTE) are important; however, the research focuses on synthetic data generation to overcome data shortage and improve classification performance. New innovations like Correlational Generative Adversarial Networks (CGANs) and SMOTE enable high-quality synthetic fraudulent cases, correcting class imbalance and increasing model generalization. This method allows fraud detection models to learn from a more diversified and representative dataset, which is important since tax fraud data are private. In contrast, the Soft-Voting Ensemble Model uses many machine learning classifiers to improve fraud detection. Ensemble learning improves performance, but high-quality synthetic data make the suggested fraud detection approach successful.

The article organizes tax fraud detection clearly and methodically. Section 2 discusses fraud detection systems' drawbacks. Section 3 introduces the Soft-Voting Ensemble Model, which uses CGANs, SMOTE, and encoder-based feature extraction. Section 4 assesses the model's accuracy, precision, recall, F1 score, and AUROC. Section 5 summarizes findings and suggests future research.

2. Literature Review

Having established the essential theory, we will now proceed to assess various state-of-the-art techniques for the detection of fraud based on machine learning. The authors [21] probed the possible complexities of solutions based on graphs, which could be used to detect fraud in the context of digital transaction data, specifically the consideration of scope, speed, and nature behind financial crime detection applications. They contend that adversarial tactics will be a significant challenge, such that it will be up to enhancing the

effectiveness of existing and upcoming graph-based solutions. Financial forgeries are growing, with a negative impact on the economy resulting from online businesses and increased internet usage. The authors posit that machine learning and data mining methodologies are currently being implemented to tackle this issue; however, improving calculation speed, data management, and recognition of unknown attack patterns is necessary.

In the paper [22], Long Short-Term Memory (LSTM) methodology was employed to propose a deep learning-based approach for the identification of financial misconduct. This model aims to enhance the efficiency of detection and current methods by making use of big data. The proposed model is evaluated on a real tax fraud dataset, and its results are compared against those from the auto-encoder model, another deep learning model, as well as other machine learning methods. LSTM performance was judged as perfect through the trial since it achieved 99.95% accuracy in under one minute. Financial deception poses a major concern adversely affecting the sustainable development of world financial markets. Although the ratio of non-fraud organizations is relatively high in comparison to fraudulent ones, this makes it all the harder to identify fraud in a dataset that happens to be highly skewed. Thus emerged some intelligent algorithms to tackle suspected financial reporting deception. Most existing methods focus on the quantitative aspect of financial statement ratios and neglect these commensurate Chinese comments.

The cited study [23] was intended to further develop a detection technology for financial misconduct that uses advanced deep learning methods to process a combination of numerical features from financial statements together with textual data derived from management comments contained in the annual reports of 5130 listed Chinese enterprises. The authors covered the gaps set by previous research by creating an elaborate system of financial indices covering all non-financial sectors previously neglected. The MD&A section's textual data were extracted using word vectors and analyzed by deep learning models together with numerical features, gaining significant advantages over traditional machine learning. Notably, the GRU and LSTM models achieved correct classification rates of 94.98 and 94.62%, respectively, on the testing samples, underscoring the effectiveness of textual features from the MD&A section to better detect financial deception.

Significant research has been done in financial crime detection using either traditional statistical methods or more contemporary machine learning techniques [24]. It is a type of artificial intelligence that applies machine learning techniques to historical data to improve the performance level in a specific task or to make correct predictions [25]. In operational domains, machine learning is used in risk mitigation, which may encompass the identification and/or prevention of threats. Machine learning is predominantly centered on identifying suspicious transactions and detecting fraud concerning operational risk, except for cybersecurity scenarios [26].

These have included, to varying degrees, bagging ensembles of classifiers such as decision trees, k-nearest neighbor, support vector machines (SVM), and Bayesian algorithms [27]. A fair conglomeration of the most pertinent research literature regarding machine learning applied to financial risk management indicates that a significant proportion of studies on the risk management tasks are primarily dependent upon machine learning [28]. Also, ref. [29] employed them for all fraud prevention. Tax fraud itself was examined using a genuine dataset and done in two phases. In the first phase, nine machine learning algorithms were developed to identify fraudulent transactions. The top three algorithms were selected for re-use in the next phase.

The CatBoost (AllKNN-CatBoost) model has been determined to be highly effective when used in conjunction with an All-K-Nearest Neighbor sampling strategy. The CatBoost with All-K-Nearest Neighbor model and its performance are compared against similar studies, with the results showing that the proposed model does outperform others with a

97.94% AUC, a 95.91% recall value, and an 87.40% F1 score. In their model, some of the authors have presented useful machine learning algorithms through which they filtered the monitor list of anti-money laundering systems [30]. A model to automate the blocked transactions verification process and assess the efficacy of various machine learning algorithms was proposed. It suggested a high-level architecture on how the machine learning aspect could be integrated into currently operating systems, and support vector machines (SVMs) were identified as superior to other algorithms. The model comprises three phases: monitoring, advising, and acting. Its goal is to enhance the efficacy and precision of watch-list filtering by using historical transaction data and additional information on transactions and blacklisted entities. The ML component can provide recommendations for blocking transactions, thus minimizing the need for human intervention. The model can easily be integrated into existing systems in the monitoring process of the filtering of watch lists.

The design of a Deep Neural Network (DNN) was done with limited features and many uninformative features for Bitcoin price prediction. The findings were successful with 53.4% accuracy and an MSE of 1.02 for the prediction [31]. Ref. [32] focused on the construction of fraud prediction models for the banking transaction space, utilizing many supervised machine learning algorithms. Their assessment of algorithmic performance utilized a dataset from 46,316 client transactions. The dataset contained 25 transaction-generated features. Features that had a 0.8 or greater Pearson correlation coefficient were discarded. The models' performance was evaluated using precision, recall, accuracy, and F1 score metrics. The outcomes showed the adeptness of machine learning models in identifying organized fraud in banking transactions. The high accuracy, recall, precision, and F1 score values clearly demonstrated how effectively the models could stymie fraud in the finance sector. Ref. [33] also used ensemble models with balancing techniques for the detection of tax fraud. The balancing techniques were evaluated as per all three classifiers in diverse combinations, with Random Forest, XGBoost, and LGBost classifiers being some of their assessments. According to the study outcome, the most effective means of tax fraud detection was balancing the dataset via SVM SMOTE and training it through the Random Forest classifier. This resulted in an F score of 0.85, with precision rated at 0.91 and recall at 0.80.

Several major scientific contributions have directly affected and supported our suggested approach, as discussed in the literature review. Analyses of graph-based fraud detection systems [21,34] indicate their potential to capture complex financial crime links but also their computational limitations. Long Short-Term Memory (LSTM) networks [22,35,36] have been shown to detect fraudulent financial transactions using deep learning. These findings justify our decision to use advanced machine learning models and encoder-based feature extraction to improve fraud detection.

In literature, the class imbalance problem, when fraudulent transactions make up a small percentage of data, was a major issue. Research has shown that SMOTE and GANs balance datasets and improve fraud detection [37–39]. Inspired by these findings, our study uses CGANs and SMOTE to generate high-quality synthetic fraud data for fraud detection model training, ensuring a more balanced and representative dataset. Ensemble learning approaches are also known to improve fraud detection categorization. Soft-Voting classifiers and other boosting and bagging methods improve forecast accuracy and reduce false positives [33]. This understanding led us to design a Soft-Voting Ensemble Model, which uses many classifiers to improve fraud detection and performance. The privacy issues and limited accessibility of real-world tax fraud datasets hamper the development of effective fraud detection models [8]. Our technique corresponds with studies supporting GAN-based synthetic data generation to create realistic yet privacy-preserving fraudulent transaction data for model training.

We base our solution on past research gaps, such as class imbalance, absence of high-quality synthetic fraud data, and machine learning model constraints. By incorporating previous research, we created an improved fraud detection framework that (1) generates synthetic fraud data using CGANs, (2) improves feature extraction using an encoder-based technique, and (3) improves classification accuracy using a Soft-Voting Ensemble Model.

3. Methodology

This research articulates a methodical approach to predict tax fraud, which begins with assembling the relevant datasets that are divided into training and testing sets. Pre-processing is performed to enhance the quality of the data. This includes the elimination of noise, anomaly detection, removal of outliers, and reducing dimensionality. SMOTE and GAN-based augmentation will be applied to add data diversity and handle class imbalance. The proposal is based on the use of Bayesian optimization to further optimize the performance of intelligent fraud detection architecture consisting of multiple models. A guarantee of reliable and precise fraud prediction is ensured as models are trained in isolation and their predictions are aggregated using a majority vote scheme. Figure 1 shows an overall structure of the proposed model.

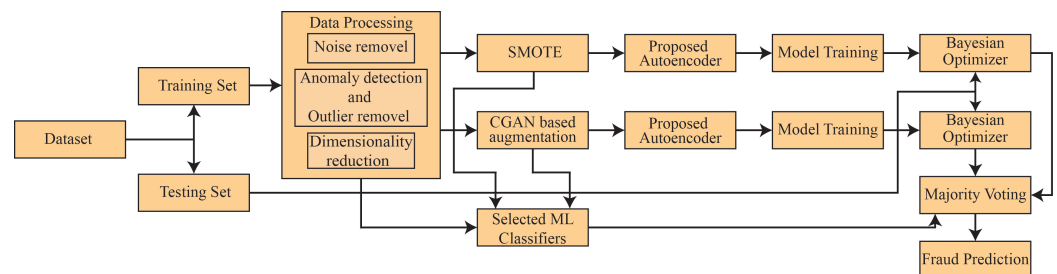


Figure 1. Overall structure of the proposed model.

3.1. Data Preprocessing

After extracting data from SET, the prime focus of the data preprocessing phase is the elimination of missing values and outliers and the selection of optimal features for training [40,41]. Such preprocessing steps are pivotal for enhancing the performance of our proposed model, as they ensure the relevance and accuracy of the data [42,43]. The feature selection phase then identifies the most valuable variables from the refined dataset, thus contributing towards the efficiency of the mode. The steps employed during the process are described below.

3.1.1. Noise Removal

The presence of noise compromises the performance of the model as it can interfere with the essential signals. To prevent this and to mitigate random data variations, we have employed several noise removal techniques like smoothing, filtering, and applied transformations [44]. Consequently, improved signal-to-noise ratio is achieved, and underlying key patterns have become more noticeable, eventually resulting in enhanced accuracy of the model. To standardize the data, Z-score normalization is applied in our proposed model. Here, the distance of each point is measured from the mean to assess its deviation from the mean. The data points with Z-scores beyond a specified threshold are considered as outliers or noise and are subsequently eliminated to refine the data and thus improve their quality and reliability.

3.1.2. Anomaly Detection and Outlier Exclusion

Anomalies in data typically point to measuring inaccuracies or revealing new findings, which may lead to inaccurate conclusions, overfitting, increased complexity, and thus skew

model accuracy. Thus, detection and prevention of anomalies is critical for accurate model predictions. In our proposed approach, we have applied the Interquartile Range (IQR) method for both detecting these anomalies and removing outliers [45,46]. The illustrations below depict a visual representation of data after applying the anomaly detection and removal technique.

3.1.3. Dimensionality Reduction

Removal of redundant and irrelevant features is needed to simplify complex datasets, thereby increasing the model's efficiency, identifying hidden patterns, and solving the issue of data sparsity. The t-SNE method is applied in the proposed algorithm to acquire effective dimensionality reduction [47]. The technique works by mapping data points from a high-dimensional space to a low-dimensional one in a non-linear manner while effectively preserving data points' similarity [48,49]. This enables visual exploration of natural patterns in the data and clustering them in two or three dimensions. Thus, a more comprehensive comprehension of the data becomes possible, which results in better understanding its distribution and characteristics [50].

t-SNE begins by measuring the similarity between data points in the original high-dimensional space using metrics such as Euclidean distance. This similarity is then modeled into a conditional probability distribution. In the resulting lower-dimensional space, t-SNE estimates initial positions for the data point and recalculates similarities, which are represented again in the form of conditional probability distributions. The algorithm then seeks to minimize the difference between the high and low-dimensional probability distributions by iteratively optimizing the data points' positions using KL divergence. It adjusts positions so that data points close in the original space remain close in the new space. The adjustments continue until the model reaches convergence in a stable low-dimensional representation.

3.1.4. Synthetic Minority Over-Sampling Technique (SMOTE)

When training classifiers, an imbalance between classes may lead to a biased model. Such a model neglects the minority class due to the dominance of the majority class. As a result, the reliability of predictive models is significantly affected, especially in domains where it is crucial to identify infrequent phenomena. To address the problem, the proposed methodology incorporates the Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic instances of the minority class [37,51]. The interpolations are created between existing minority class instances. New examples are then integrated into the training dataset. Initially, SMOTE selects a random minority class sample x_i and determines its k -nearest neighbors. A random neighbor $x_{i,k}$ is then chosen from the neighbors set. Next, a new synthetic instance, x_{new} , is created on the line segment connecting x_i and $x_{i,k}$ by employing the following equation:

$$x_{new} = x_i + \gamma(x_{i,k} - x_i) \quad (1)$$

with γ chosen randomly from the range 0 to 1. This strategy ensures that newly generated samples reflect the minority class traits accurately, thus enhancing the classifier's ability to effectively identify the minority class instances. The integration of SMOTE has enabled us to achieve a balanced class distribution during the model's training process, which has greatly improved the classifier's performance towards the minority class samples. Apart from accuracy, other metrics such as precision and recall for the target class have also improved. This shows that SMOTE addresses the class imbalance issue quite well.

3.2. Correlational Generative Adversarial Network (CGAN)

While deep learning (DL) models have demonstrated remarkable classification capabilities across various domains to exhibit optimal performance, they require large, varied datasets. Smaller dataset generation is a common challenge in the field of wireless communications owing to the temporal correlation traits of wireless signals, which complicates the process of data collection, thus resulting in datasets that are not true representations of real-world scenarios. The generator learns to make synthetic data that mimic genuine samples, and the discriminator tries to discriminate between real and generated data in a vanilla GAN. Vanilla GANs have low-diversity outputs due to mode collapse and feature distribution control issues. We present a Correlational Generative Adversarial Network (CGAN) with contextual attribute conditional dependencies for produced fraud instances. CGAN preserves tax fraud correlations in simulated fraud instances using conditional feature guidance. This update enhances data quality, diversity, and congruence with real-world fraudulent trends, making it better for fraud detection.

Auxiliary Classifier GAN [52] (ACGAN) adds an additional classifier to vanilla GANs to predict sample class labels. Improves synthetic data label quality but may not be ideal for tax fraud detection, where feature correlations and transaction patterns are more important than clear class labels. Conditional Tabular GAN (CTGAN) [53,54] is designed for organized tabular data and handles mixed categorical and continuous features well. Mode-specific normalization makes it ideal for organized financial fraud detection. ACGAN improves class-based synthetic data creation but not for transaction-based fraud patterns since it focuses on categorization rather than feature correlation modeling. CTGAN is good for structured data but may miss conditional fraud dependencies between features. Our proposed CGAN model explicitly conditions synthetic data on key fraud indicators to construct fraudulent transactions with realistic financial attribute dependencies. To overcome this limitation, we have developed a Correlational Generative Adversarial Network (CGAN) model for data augmentation. Figure 2 illustrates the overall structure of our proposed GAN model, which aims to enhance the dataset size by synthetically generating data samples that mimic the real-world characteristics of data, thus enhancing the dataset for effective training of the DL classifiers [55,56].

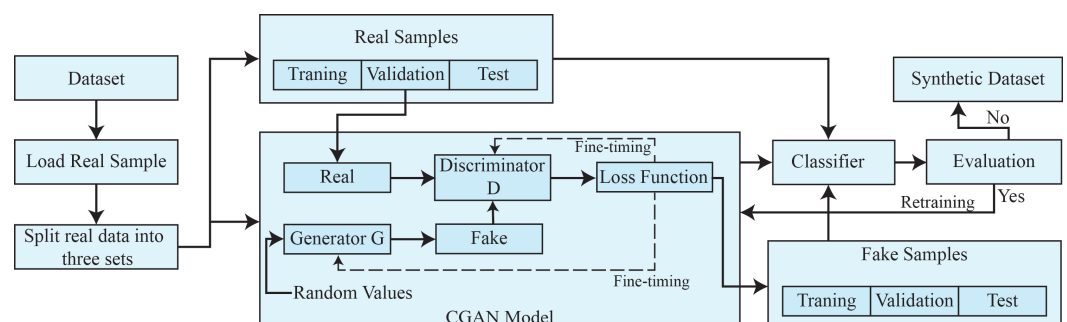


Figure 2. Structure of proposed CGAN model to enhance the dataset size synthetically.

The GAN architecture is built around two neural networks: the generator G_{GAN} and the discriminator D_{GAN} . The generator synthesizes fake signals by learning and imitating the distribution of real data from a noise input z based on a noise model distribution P_z (the generated output is expressed as $G_{GAN}(z)$). Unlike traditional models that only memorize input–output pairs, the generator is specifically designed to learn and replicate the data distributions within actual datasets. Conversely, the discriminator acts as a binary classifier to differentiate between the synthetic data $G_{GAN}(z)$ produced by the generator and the real data from the dataset, which are adhering to the distribution P_{data} , assigning a value of 0 to synthetic data and 1 to real data.

The proposed model employs two different loss functions for the generator and discriminator, thus optimizing the performance of both distinctly. These loss functions are derived from the comparative analysis of two distinct probability distributions: P_z (output by the generator) and P_{data} (output by the real dataset). The functionality of GAN is designed based on a min–max game. The generator aims at minimizing its loss, thereby generating synthetic samples that mirror the real samples, while the discriminator is focused on maximizing the function to accurately distinguish between generated and real data samples. The formula that directs the learning process between the generator and the discriminator in our proposed model is expressed as

$$E_z = \mathbb{E}_{z \sim P_z(z)} [\log(1 - D_{GAN}(G_{GAN}(z)))] \quad (2)$$

$$E_r = \mathbb{E}_{r \sim P_{data}(r)} [\log(D_{GAN}(r))] \quad (3)$$

$$\min_{G_{GAN}} \max_{D_{GAN}} V(D_{GAN}, G_{GAN}) = E_r + E_z \quad (4)$$

where E_r denotes the expected value derived from all samples in the real dataset, $D_{GAN}(r)$ indicates the discriminator's estimation of the likelihood that these samples are real, and E_z represents the expected value over the generated outputs $G_{GAN}(z)$ and $D_{GAN}(G_{GAN}(z))$, showing the likelihood of these generated samples being classified as real. The formula involves two components: $\mathbb{E}_{r \sim P_{data}(r)} [\log(D_{GAN}(r))]$ and $\mathbb{E}_{z \sim P_z(z)} [\log(1 - D_{GAN}(G_{GAN}(z)))]$. The former assesses the discriminator's ability to recognize real data, whereas the latter measures its capability to recognize the synthetic data.

The proposed CGAN loss functions aim to maintain feature correlations in fraudulent transactions while producing high-quality synthetic data. Our method adds conditional dependencies to both the generator and discriminator, unlike vanilla GANs that only optimize adversarial losses. The generator loss is defined as

$$\mathcal{L}_G = \mathbb{E}_{z \sim p_z(z)} [-\log D(G(z|c))] \quad (5)$$

The generated fraud sample $G(z|c)$ is based on fraud-related attributes c , guaranteeing that synthetic transactions mimic real fraud patterns. The Discriminator Loss is defined as

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{data}(x)} [\log D(x|c)] - \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|c)))] \quad (6)$$

where $D(x|c)$ indicates the likelihood of a transaction being legitimate based on its fraud-related characteristics. This preserves feature correlations and helps the discriminator distinguish actual from manufactured fraud situations.

Despite their advantages, GANs are known to encounter the challenge of non-convergence due to factors such as Nash equilibrium. We have addressed the issue by adopting Auxiliary Classifier GAN (ACGAN), which is a method to improve the training of GAN. This approach enables the model to not only distinguish real data from synthetic ones but also classify the input into classes. Apart from this, a classifier (C) is added with a similar configuration to that of the discriminator. The classifier is trained when the Nash equilibrium state is reached to ensure the synthetic samples produced by the GAN closely match the distribution of the real dataset. After rigorous evaluation, the synthetic samples that are most convergent and thus indistinguishable from the real ones are collected into a synthetic dataset. This dataset is essential for the training of our proposed model, as it significantly boosts the precision and robustness of our authentication processes.

For transparency and reproducibility, we included Table 1 with a sample of the raw dataset and the final processed dataset after preprocessing and augmentation. This table shows how data are modified across the pipeline using valid and fraudulent activities.

Real-world tax records, including reported income, declared deductions, tax refunds, filing consistency, and audit flag status, are included in the raw dataset. Due to increased deductions and tax refunds and inconsistent filing behavior, fraudulent transactions are often detected. Fraud classification is improved via feature engineering, anomaly detection, and synthetic data augmentation in the processed dataset. The encoder-based feature extraction technique calculates the anomaly score, which identifies odd transactions based on learned patterns to aid classification. To balance the dataset and increase fraud detection, CGAN and SMOTE synthetic records are added.

Table 1. Sample of raw and processed legitimate and fraudulent transaction dataset.

Transaction ID	Reported Income (\$)	Declared Deductions (\$)	Tax Refund (\$)	Filing Consistency	Audit Flag	Anomaly Score	Final Fraud Label
Raw Dataset (Before Processing)							
1001	55,000	10,000	1200	Consistent	No	-	Legitimate
1002	75,000	35,000	8500	Inconsistent	Yes	-	Fraud
1003	45,000	5000	900	Consistent	No	-	Legitimate
1004	120,000	60,000	15,000	Inconsistent	Yes	-	Fraud
Final Dataset (After Preprocessing and Augmentation)							
2001 (Synthetic)	68,000	28,000	6200	Consistent	No	0.15	Legitimate
2002 (Synthetic)	82,500	40,000	10,200	Inconsistent	Yes	0.87	Fraud
2003	50,500	6800	1100	Consistent	No	0.10	Legitimate
2004	110,000	52,000	13,500	Inconsistent	Yes	0.92	Fraud

3.3. Proposed Autoencoder

The proposed autoencoder is demonstrated in this section and is illustrated in Figure 3. Autoencoder, a type of artificial neural network to perform unsupervised learning, is employed in the proposed work [57,58]. An autoencoder employs an encoder function to modify the inputs and decoder functions to regenerate the input data from the encoded one. The process begins by initializing the model parameters, including the weights and biases for both the encoder and decoder. The parameters are then employed to process the n_f -dimensional input features from the feature vector X in the input layer of the autoencoder. The input layer takes in data point X_i and transforms it into a lower-dimensional representation Z_H (often called the latent space) using

$$Z_H = \sigma_{En}(\theta_{En}X_I + \beta_{En}) \quad (7)$$

where σ_{En} is the activation function (such as the sigmoid function) at the encoder to add non-linearity, and θ_{En} and β_{En} are weights and biases in the encoder [59,60]. The decoder then converts Z_H into a reconstructed output prediction Y as

$$Y = \sigma_{Dc}(\theta_{Dc}Z_H + \beta_{Dc}) \quad (8)$$

where Y is the prediction of X_i , σ_{Dc} is the activation function at the decoder, and θ_{Dc} and β_{Dc} are weights and biases in the decoder. To measure the reconstructed error, a square error cost function is used as it penalizes larger errors heavily, thus enabling the model to significantly reduce such errors during training. The squared error cost function, $J(X_I, Y)$, is computed as

$$J(X_I, Y) = |X_I - Y|^2 \quad (9)$$

To further refine the features, output Z_H of the hidden layer is passed through several fully connected layers to obtain the new output X_F as

$$X_F = \sigma_F(\theta_F Z_H + \beta_F) \quad (10)$$

where σ_F is the activation function at the fully connected layers, θ_F and β_F are weights and biases, respectively. The proposed methodology involves a differentiable forest model, known as a neural decision forest [61,62], in contrast to traditional machine learning models like the random forest, which lack differentiability and hence are unable to be utilized in end-to-end neural network training. By employing this model, each node is assigned a probabilistic distribution, which enables differentiating each decision tree.

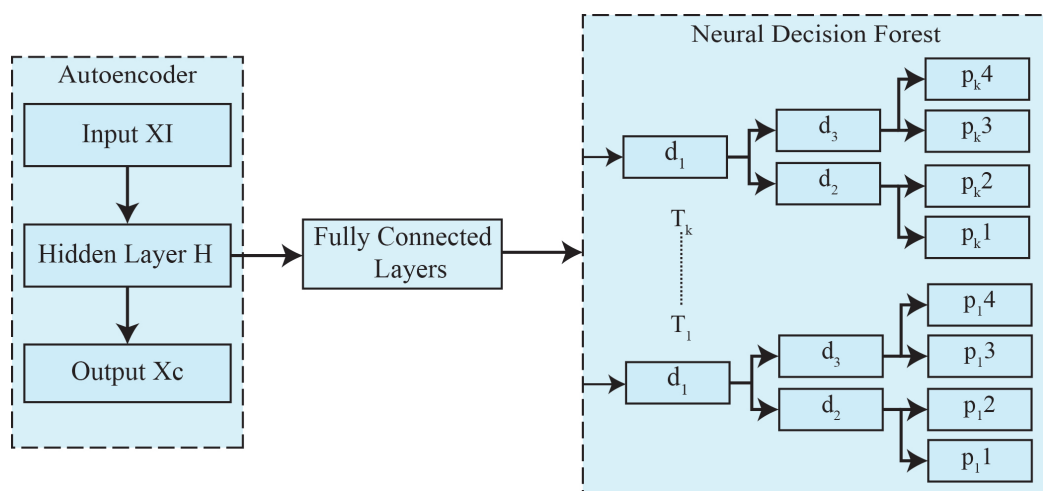


Figure 3. Architecture of proposed autoencoder having fully connected and neural decision forest blocks.

Consider an architecture with m decision trees that construct the neural decision forest, with each tree containing T_D decision nodes and T_L leaf nodes. Leaf nodes are the endpoints of the tree and a probability distribution. P_{leaf} over Y is associated with each T_L node that enhances the decision-making process. There exists a decision function $D_{f,i}(X_F; \Phi)$ for each $T_D(i) \in T_D$ that ranges from 0 to 1. The output of the function indicates the likelihood that the input that has arrived at $T_D(i)$ will proceed to the left subtree. The function for the i^{th} decision node is expressed as

$$D_{f,i}(X_F; \Phi) = \sigma_{ND}(f_{d,i}(X_F)) \quad (11)$$

where σ_{ND} is the activation function for the neural decision environment, the set of all the model parameters, $f_{d,i}(X_F) = W_F X_F$, is the linear transformation function for the underlying decision node, and W_F is the weight matrix that is applied to input features.

Every tree in our model follows a traditional binary tree architecture, i.e., every node splits into two smaller trees. For instance, a tree with a depth of 1 has two decision nodes at the first level and four leaf nodes overall. Generally, the formula to determine the number of decision nodes for a tree of depth d would be 2^d , and that of leaf nodes would be 2^{d+1} . Thus, the following mathematical relation will be used to compute the probability that a sample in tree k belongs to class Y :

$$P_{Tk}[y | X_F, \Phi, P_{leaf}] = \sum_{l \in T_L} \left(\prod_{T_D(i) \in T_D} (D_{f,i}(X_F; \Phi)_{left} \cdot (1 - D_{f,i}(X_F; \Phi)_{right})) \right) \quad (12)$$

Within each decision node i , the decision function $D_{f,i}$ computes the probability of directing the input towards the left, mentioned as $D_{f,i}(X_F; \Phi)_{left}$ and the probability $1 - D_{f,i}(X_F; \Phi)_{right}$ for not directing towards the right path.

To reach any leaf node, such as T_{l1} in the first tree, D_{l1} , decisions are taken at nodes as T_{d1} and T_{d2} , with the probability of reaching T_{l1} as a product of probabilities at both decision nodes ($T_{d1} \cdot T_{d2}$). Similarly, the probability of reaching the other leaf nodes can be computed using the same approach. Let $P_{leaf}(T_{l1})$ and $P_{leaf}(T_{l2})$ be the probabilities at nodes T_{l1} and T_{l2} , respectively, for classifying a sample as Y (tax-compliance status), then $\vec{P}_y(T_{l1}, T_{l2}) = (P(T_{l1}), P(T_{l2}))$. Here, $P(T_{l1})$ and $P(T_{l2})$ denote the probabilities that the samples are tax-compliant.

Each tree T_i from the tree ensemble $T_{Ensemble} = \{T_1, T_2, \dots, T_m\}$ generates a prediction for the sample x . These individual predictions are then averaged to yield a cumulative prediction as

$$P_E[y | x] = \frac{1}{M} \sum_{m=1}^M (P_{leaf}(T_m)[y | x]) \quad (13)$$

The final predicted class $\hat{y}$ of the model is the one that has the highest collective probability from all the trees in the ensemble as

$$\hat{y} = \arg \max_y P_E[y | x] \quad (14)$$

The overall loss $J(T_{Ensemble})$ for the neural decision forest is computed by averaging all the losses of individual trees in the ensemble as

$$J(T_{Ensemble})(x, y, \Phi, P_{leaf}) = \mathbb{E}_{x \in X} \left(\mathbb{E}_{T \in T_{Ensemble}} \left(J_T(x, y, \Phi, P_{leaf}) \right) \right) \quad (15)$$

And for each individual tree in the ensemble, the loss function $J_T(x, y, \Phi, P_{leaf})$ for this classification problem is defined as

$$J_T(x, y, \Phi, P_{leaf}) = - \left(y \cdot \log P_T[y | x, \Phi, P_{leaf}] + (1 - y) \cdot \log(1 - P_T[y | x, \Phi, P_{leaf}]) \right) \quad (16)$$

$$= - \log P_T[y | x, \Phi, P_{leaf}] \quad (17)$$

The final simplified version of the mathematical relation represents the binary cross-entropy loss. The objective is to minimize the overall loss, such as

$$J(X, y, \Phi, P_{leaf}) = \arg \max_{\Phi} \left(J_{En, De}(X_I, Y) + J(T_{Ensemble})(x, y, \Phi, P_{leaf}) \right) \quad (18)$$

In the proposed technique, RMSProp [63] is employed for parameter optimization, which is a refined version of RProp specifically for mini-batch updates. Furthermore, RMSProp employs not only the direction of the gradients for updates but also considers magnitudes as well. To modify the learning rate of a parameter, it is divided by the running average of the recent gradients' magnitude. The updates to the parameters are made as follows:

$$\nabla_t = \nabla J(T_{Ensemble})(\Phi_{t-1}; B, P_{leaf}) \quad (19)$$

$$V_t = V_{t-1} + \nabla_t \otimes \nabla_t \quad (20)$$

$$\Phi_t = \Phi_{t-1} - \frac{\alpha}{\sqrt{V_t} + \delta} \times \nabla_t \quad (21)$$

where ∇_t is the gradient of the loss function at time t , B denotes the randomly selected mini-batch, V_t represents the accumulated squared gradient, α is the learning rate, and a small constant δ is added to prevent division by zero. To compute the probability

distribution $P_{leaf}(T_l)$ at the leaf node, the proposed methodology integrates the softmax function with the RProp technique to refine the update process. The entire process is carried out as follows:

$$\nabla_t^\lambda = \nabla J(T_{Ensemble})(\lambda_{t-1}; B, \Phi) = \nabla J_T(\lambda_{t-1}; x, y, \Phi) \quad (22)$$

$$V_t^\lambda = V_t^\lambda + \nabla_t^\lambda \otimes \nabla_t^\lambda \quad (23)$$

$$\lambda_t = \lambda_{t-1} - \frac{\alpha^\lambda}{\sqrt{V_t^\lambda + \delta}} \otimes \nabla_t^\lambda \quad (24)$$

$$\lambda_t = \text{softmax}(\lambda_t) \quad (25)$$

Algorithms 1 and 2 present the pseudo-code for both the training and testing processes of the proposed autoencoder model.

Algorithm 1: Training the proposed Autoencoder model

Input: Feature vector X_I for a taxpayer record, label y for the record being tax compliant or tax evader, epochs count n_{epochs} , decision tree count n_{trees} , decision tree depth n_{depth} , batch count n_{batch}

Output: Probability P for label y

1. Construct a neural decision forest with n_{trees} number of trees and each tree having depth equal to n_{depth} .
 2. Randomly initialize the values for Φ and P .
 3. Shuffle the dataset randomly to prevent order bias during training.
 4. **for each epoch ranging from** $i = 1$ **to** n_{epochs} **do**
 - (a) Divide the dataset into n_{batch} .
 - (b) **for each batch ranging from** $i = 1$ **to** n_{batch} **do**
 - i. Apply RProp optimization to update Φ .
 - end**
 - (c) Apply Equations (20) to (23) to update P .
- end**
-

The autoencoder is used for feature extraction and dimensionality reduction during model training to retain the most informative latent representations and eliminate redundant information. However, t-SNE is utilized as a data visualization post-processing tool to assess and explain how closely synthetic fraud data matches genuine fraud incidents in a lower-dimensional space.

Algorithm 2: Prediction process using the proposed Autoencoder model**Input:** Feature vector X_I for a taxpayer record**Output:** Prediction $\hat{y}$ for the tax record

1. Transform input features into a lower-dimensional representation as $Z_H = \sigma_E(\theta_E X_I + \beta_E)$.
2. Convert Z_H into a reconstructed output as $Y = \sigma_D(\theta_D Z_H + \beta_D)$.
3. **for each possible class i (tax compliant or tax evader) do**
 - (a) Initialize $p(y = i) = 0$.
 - (b) **for each tree k from 1 to K do**
 - i. $p = P_k[y = i \mid X_F, \Phi, P_{leaf}]$.
 - ii. $p(y = i) = p(y = i) + p$.
 - end**
 - (c) $p(y = i) = \frac{1}{K} p(y = i)$.
- end**
4. $\hat{y} = \max_i p(y = i)$.
5. **return** $\hat{y}$.

3.4. Hyperparameter Optimization

The parameter selection for the CNN model plays a crucial role as it impacts the learning rate of the model and enables effective modeling of the data patterns. At the core, CNN model training is based on the following fundamental parameters:

- Activation functions that are critical in determining the complex patterns in data;
- Batch size defining the training examples count that would be used to train the model in one iteration;
- Dropout rate specifying a range so that some of the neurons can be nullified from contributing towards the next layer;
- Filter size to set filter dimensions in each convolution layer;
- Number of filters that convolve with the data in each convolution layer to extract distinct data features;
- Number of convolutions, where each convolution is a mathematical operation in which the kernel slides across the input data to extract meaningful patterns from it;
- Epoch count where each epoch refers to one complete cycle of model training;
- Validation frequency determines how frequently to validate the model during training;
- Stride defines the step size by which the filter should move while performing a convolution operation;
- Max pooling applied on the convolution layer for dimensionality reduction by retaining important features.

3.4.1. Encoder Architecture Optimization

Several hyperparameter optimization techniques exist that can be employed in the tuning process of the CNN model. These include Grad Student Descent (GSD), Grid Search (GS), Random Search (RS), Gradient Descent (GD), and Bayesian optimization [64,65]. GSD is a basic manual tuning method that relies significantly on trial and error and is often unsuitable for complex models with hyperparameters involving non-linear relations. GS, on the other hand, is known for its exhaustive search within a predefined grid as it methodically investigates every possible combination within that grid. However, it falls short of reaching the global optimum of the objective function [66].

In the RS approach, various hyperparameter values are randomly chosen within pre-defined bounds and are trained to meet a particular objective function. This technique does not build on the successes of previous iterations, often leading to inefficient evaluations of the objective function. However, RS does not leverage information gained from previously successful hyperparameter regions, thus leading to needless objective function evaluations. In contrast, GD is widely employed for fine-tuning continuous hyperparameters by determining their gradients. While it effectively finds local minima in convex functions, GD struggles to search for global minima in non-convex functions [67]. Given the challenges mentioned, there is a need for more sophisticated methods such as Bayesian optimization (BO) for optimal selection of the hyperparameters [68]. BO operates through an iterative process and determines the future evaluation points by analyzing the results of previous iterations. This approach employs a surrogate model and an acquisition function to effectively choose hyperparameter configurations. A detailed description of the BO process is provided in the next section.

3.4.2. Bayesian Optimization

BO is an effective statistical method used to optimize resource-intensive objective functions. It leverages Bayes' Theorem to update the objective function's probability model [69]. To achieve this, BO incorporates both pre-existing knowledge and new data obtained through an iterative evaluation process, thus resulting in the efficient tuning of hyperparameters for the machine learning models. Let c denote a specific hyperparameter value, and C_{BO} describes the configuration space associated with c ; then the mathematical relation for Bayes' Theorem can be illustrated as

$$P_{BO}(c | C_{BO}) = \frac{P_{BO}(C_{BO} | c)p(c)}{P_{BO}(C_{BO})} \quad (26)$$

where $P_{BO}(c | C_{BO})$ represents the posterior probability of hyperparameter c within configuration space C_{BO} , $P_{BO}(C_{BO} | c)$ depicts the likelihood of occurrence of C_{BO} given c , and $P_{BO}(c)$ refers to the initial probability of c .

In BO, Gaussian Process (GP) often serves as the preferred surrogate model for effective modeling of objective functions. Within the structure of the Bayesian Optimization-Gaussian Process (BO-GP) framework, the GP operates with a function f defined by a mean μ and covariance σ^2 to estimate values as a normal distribution N , as specified:

$$P_{BO}(y | c, C_{BO}) = N(y | \mu, \sigma^2) \quad (27)$$

where $y = f(c)$ represents the outcomes for each hyperparameter value c . Following each objective function evaluation, the BO refines the surrogate model using the latest estimation of $f(c)$. Additionally, an acquisition function $h(c)$ that is based on GP is utilized to refine the search space by identifying the next point c for evaluation. This process enables obtaining the optimal hyperparameter c_{opt} that minimizes the loss and maximizes validation accuracy, as

$$c_{opt} = \arg \min_{c \in C_{BO}} f(c) \quad (28)$$

Bayesian optimization optimizes the Soft-Voting Ensemble Model, CGAN, and encoder-based feature extraction hyperparameters in our proposed model. To enhance the performance of the proposed Soft-Voting Ensemble Model, CGAN, and Encoder-Based Feature Extraction Mechanism, we employ BO for hyperparameter tuning. The optimization process explores key parameters that influence model accuracy, convergence speed, and overall robustness. For the Soft-Voting Ensemble Model, BO fine-tunes parameters such as the learning rate of individual classifiers (optimal: 0.01), the number of estimators

in boosting models (optimal: 100), the max depth of decision trees (optimal: 6), and dynamic voting weights for classifiers. These optimizations ensure that the ensemble model effectively balances precision and recall, improving fraud classification performance. For CGAN, BO refines the generator and discriminator learning rates (both set at 0.0002), batch size (optimal: 64), and latent space dimension (optimal: 128) to ensure the generation of high-quality synthetic fraudulent transactions while avoiding mode collapse. These parameters allow CGAN to create synthetic fraud cases that maintain realistic correlations, strengthening the model's ability to generalize. In the Encoder-Based Feature Extraction Mechanism, BO is applied to determine the number of hidden layers (optimal: 3), the function (optimal: ReLU), and the dropout rate (optimal: 0.3). These optimizations help the encoder learn the most informative latent representations while reducing overfitting.

3.5. Majority Voting Ensemble Model

To ensure high prediction accuracy, we have merged individual predictions from multiple base classifiers (BCs) by employing the weighted majority voting rule. Each classifier, denoted as C_r where $r = 1$ to R , is assessed on a predetermined validation dataset D_v and achieves an accuracy of A_r . Every test instance x_t , where $t = 1$ to T , is processed by providing its respective feature vector to all the classifiers within our ensemble framework. Let w_r denote the weight assigned to each C_r , x_t be the feature vector being evaluated, $C_r(x_t)$ depict the individual prediction of each C_r , and $\zeta(\cdot)$ be the Kronecker delta function; the individual predictions acquired by all classifiers are then aggregated based on each classifier's achieved accuracy as

$$\tilde{y} = \arg \max \sum_{r=1}^R w_r r \zeta(C_r(\tilde{x}_t, l)) \quad (29)$$

where w is calculated as $w = w_r = \frac{A_r}{\sum_{r=1}^R A_r}$, and the Kronecker delta function rewards accurate predictions with a value of 1 and penalizes incorrect predictions with a value of 0 as

$$\zeta(C_r(\tilde{X}_t), l) = \begin{cases} 1, & C_r(\tilde{X}_t) = l \\ 0, & C_r(\tilde{X}_t) \neq l \end{cases} \quad (30)$$

This process is repeated across all test instances in dataset D_e , culminating in the final prediction set $\{\tilde{Y}_t\}_{t=1}^T$. For this study, along with the proposed optimized encoder architecture, the following 5 classical BCs are selected.

3.5.1. Multilayer Perceptron (MLP)

MLPs are feedforward artificial neural networks designed to create a series of outputs from a given set of inputs. Although the term MLP has occasionally come to refer generally to any feedforward artificial neural network, it is most specifically associated with networks that involve some combination of multiple layers of perceptrons. An MLP is characterized by the interconnections among several layers of input nodes in a directed graph containing input and output layers. Such connections enable the MLP to track complex patterns and relationships in the data, making it an extremely powerful tool for a host of machine learning tasks, including regression and classification.

3.5.2. Stochastic Gradient Descent (SGD)

SGD is a widely used optimization technique in machine learning applications for determining model parameters that align predicted outputs with actual outputs. Stochastic refers to the method of random selection of the data necessary by the algorithm. SGD modifies the parameters of the model iteratively so it samples from a very small subset of data at each

iteration, unlike standard gradient descent, which calculates the gradients based on the entire dataset. Because of this random selection, SGD is less memory-dependent and faster—for big datasets—with time-proven convergence toward its objective function through smoothing.

3.5.3. Adaptive Boosting (AdaBoost)

Boosting is an ensemble learning method that combines several weak classifiers into a strong classifier. The process consists of adding another model after the previous one to learn from the mistake made by it. This continues until the training data are perfectly predicted or the limit on the number of rounds has been reached. One of the most famous and oldest methods of boosting is AdaBoost, which stands for Adaptive Boosting. It is a binary classification algorithm designed to improve the prediction performance of a system with each iteration. This algorithm works by combining weak classifiers adaptively to form a single strong classifier. Due to its simplicity and efficiency, this boosting algorithm has become the basic building block of boosting strategies.

3.5.4. Extreme Gradient Boosting (XGBoost)

XGBoost is a highly efficient and scalable machine learning algorithm widely used in distinct domains to accomplish state-of-the-art results during data competitions. This technique is tree-based boosting, which takes a sequence of weak classifiers with relatively low accuracy to create a strong classifier with a level of performance that is very much improved. This was designed specifically for that purpose. XGBoost is a preferred algorithm for competitive machine learning, as well as real-world projects, owing to its richness in efficiently optimizing classification tasks through iterative refinement of predictions and minimum errors.

3.5.5. Random Forest (RF)

Random Forest (RF) is a machine learning classifier that works quite well for theoretical models or practical use. Randomly selected independent variables (features) make subsets of the data along with row sampling. Hyperparameter optimization is conducted to find the depth and geometry of trees used for modeling in the forest. In classification tasks, the most common prediction from all individual DTs in the forest is chosen as the final output by majority voting. The random selection of root nodes and splits in RF injects chance dimensions to ensure robustness and to avoid overfitting. This distinctly separates RF from the conventional DT algorithms. In particular, Random Forest is effective in processing complex datasets and increasing model accuracy further.

4. Experimental Results and Discussion

This section provides an explanation of how the proposed model functions when applied to the dataset used for detecting tax fraud to identify fraudulent transactions.

4.1. Dataset Description

Table 2 shows the collected total instances from 2017 to 2021 after data acquisition is completed. Total instances collected are 3852; out of these, 3105 instances contain complete data, while there are 747 with missing information-completed data. Out of the completed data, 430 are given the label “1” for legitimate instances while 2675 instances are given the label “0” for fraudulent instances. This means that around one-sixth of the data collected has been labeled; therefore, it is safe to say that the data collected are imbalanced. Then data filtering is done to eliminate values that seem unreasonable. In our case, an unreasonable value is associated with any value that does not accurately reflect reality, for example, negative inventory attributes or the presence of incomplete values. Thus, the total instances left for further analysis are 2942. To get transference, D is halved, whereby the total number of

instances for “Data Augmentation” becomes 1471. As a rule of thumb, enough instances to build basic models would normally approximate 1000 instances; however, for more complex models, such as ANNs or deep learning, a greater number of training data are required. Thus, this research will show the effectiveness of data synthesis to provide the analysis with more instances. Some observations were made after the acquisition of the dataset. One must review closely the units provided in many financial statements. Some firms report some information in various currencies: that is, USD, thousand USD, and million USD.

Table 2. Year-wise statistics of dataset.

Year	Missing	Completed	Yearly Total
2017	150	563	713
2018	148	587	735
2019	146	619	765
2020	154	647	801
2021	149	689	838
Total	747	3105	3852

Data from numerous years of tax-related financial records were used in this study. In Table 2, the manuscript presents year-wise statistics on the dataset, revealing 3852 cases with 3105 complete records and 747 missing information. Financial and transactional features important to tax fraud detection are included in the classification dataset. The dataset classifies transactions as legitimate (1 or 0). Dimensionality reduction and anomaly detection removed unnecessary features after preprocessing. Key categorization attributes include (a) financial data (income, expenses, deductions, tax refunds); (b) taxpayer behavior (e.g., tax filing consistency, income differences across years); and (c) anomalies and inconsistencies (excessive deductions, tax report discrepancies). Synthetic data augmentation employing SMOTE and CGANs was used to correct class imbalance, providing more samples for fraudulent instances. The resulting dataset was used to train a Soft-Voting Ensemble Model that detects fraud using multiple machine learning classifiers.

In our study, we take a 250-size sample and train the CTGAN for 40,000 epochs to produce synthetic datasets. The selected dataset consists of 15,000 synthetic instances produced by SMOTE and CTGAN. The quality of the synthesis is evaluated using the Quality Score (QS) of the synthetic datasets obtained. Table 3 shows a summary of QS for each augmented dataset.

The QS in Table 3 is introduced as a composite metric designed to evaluate the effectiveness of fraud detection models by considering both classification performance and data credibility. This metric is particularly relevant in synthetic data-driven fraud detection, where model performance must be balanced against the reliability of the generated data. The QS is computed using the following formula:

$$QS = (\alpha \times F1) + \beta \times DC \quad (31)$$

The harmonic mean of precision and recall, F1, shows the model’s capacity to detect fraud while limiting false positives. Data credibility (DC) is assessed using statistical similarity measurements (t-SNE visualizations) to ensure synthetic fraud data closely resembles real fraud transactions. Adjust α and β weighting coefficients based on application needs ($\alpha = 0.7$, $\beta = 0.3$) to prioritize model performance and data dependability.

Table 3. Summary of QS against each augmented dataset.

Method	Generated Samples	QS
SMOTE	15,000	0.94
CTGAN	15,000	0.91

4.2. Performance Metrics

The current studies underline Class A, class of fraud because of its enormous misclassification cost. Efficacy of the proposed model is evaluated using several performance metrics. Due to its high degree of bias, accuracy alone cannot gauge model performance in the concerned dataset. Therefore, F1 score, recall, and precision are used along with accuracy. Additionally, the model's ability to differentiate between the two classes of data observations is quantified by the Area Under the Receiver Operating Characteristic (AUROC) curve. The recall is given preference during the implementation, as it reflects the classifier's sensitive ability to identify fraud transactions. So, any average classifier may score around 0.50, while the best classifier can score 1.00. Recall for the fraudulent class shall fall anywhere between 0.50 and 1.00. Any value of recall over 0.80 is generally regarded as a good classifier. This connotes that the classifier has made good predictions by classifying fraudulent transactions against non-fraudulent ones. The computation for misspecification of fraudulent classes using False Negative Rate (FNR) and False Positive Rate (FPR) was also looked upon. In terms of performance measurement, the misclassification rate refers to the number of false predictions, without distinguishing between positive and negative. It is crucial to predict fraud transactions accurately to prevent credit card fraud. Hence, in this analysis, the model achieving the highest recall and the lowest false-negative rate is selected, meaning the number of fraudulent transactions classified as legitimate transactions.

4.3. Comparative Analysis

In the present study, a series of ML algorithms are applied to the original (highly imbalanced) tax fraud dataset. After that, SMOTE and CGAN will be considered for augmenting the tax fraud dataset. The application of two sampling techniques, indicated in Table 3, results in the generation of three slightly different tax fraud datasets.

The performance of six classifiers is compared in Table 4 across three datasets: original, SMOTE, and CGAN. The performance of all classifiers is enhanced by SMOTE and CGAN, which is indicative of the influence of sampling techniques on classification outcomes. The F1 score of MLP goes up from 70.01% on the original dataset to 82.30% on CGAN; its accuracy rises from 70.94% to 82.43%, and its FNR goes down from 1.77% to 0.91%. All classifiers see such increased performance; other trends played out similarly for SGD, AdaBoost, XGBoost, and RF. All classifiers maintained upward shifts in F1 scores, precision, recall, and accuracy from original to CGAN datasets, with significant reductions in FNR and FPR. This shows that these sampling techniques were relevant in improving predictive accuracy and reducing misclassification errors.

A baseline classifier classifying all transactions as legitimate would obtain 86% accuracy in the original dataset, which comprises 86% legitimate cases and 14% fraudulent ones. Table 5 shows models with lesser accuracies, such as 71% (MLP) and 76% (AdaBoost) on the original dataset, due to many variables. Models like MLP and AdaBoost are more sensitive to class imbalance. Instead of favoring the majority class, these algorithms optimize their predictions across both classes, misclassifying some legitimate cases to increase fraud detection. This method reduces accuracy but improves fraud detection. Second, the bias-variance trade-off affects MLP generalization on imbalanced datasets without proper management, which increases misclassifications. Noise and feature distribution restrictions in the dataset also contribute. Patterns that do not generalize well may cause

inaccurate classifications in some models. Models that do not use all information or catch fraud patterns may also underperform. Finally, training data constraints like insufficient fraud-related incidents affect model learning and classification. F1 score, recall, and AU-ROC provide a more complete picture of fraud detection algorithms' performance than accuracy alone. Some models' poorer accuracy shows they prioritize fraud detection above majority-class categorization, which is essential for tax fraud detection. The study uses synthetic data augmentation (CGAN & SMOTE) and ensemble learning to improve fraud detection and model robustness.

Table 4. Performance of five classifiers along with the proposed encoder model across three datasets.

Classifier	Dataset	F1 Score	Precision	Accuracy	Misclassification Rate		Detection Rate	
					AUROC	Recall	FPR	FNR
MLP	Original	70.01	73.5	70.94	4.53	71.84	71.13	1.77
	SMOTE	81.61	88.91	88.88	9.73	87.45	88.63	0.85
	CGAN	82.30	91.59	82.43	4.25	87.57	87.54	0.91
SGD	Original	81.24	81.28	81.11	2.56	84.24	80.97	1.98
	SMOTE	90.43	80.80	85.56	7.57	84.05	87.94	0.93
	CGAN	91.75	93.83	94.80	9.39	85.31	83.96	1.38
AdaBoost	Original	77.24	73.03	76.12	9.20	78.76	70.88	1.93
	SMOTE	80.32	90.62	91.84	1.14	88.32	84.29	0.89
	CGAN	88.86	83.88	85.20	6.80	86.41	90.20	1.64
XGBoost	Original	81.93	85.36	81.95	4.66	86.01	87.82	0.77
	SMOTE	88.62	83.37	91.33	8.34	80.85	91.55	1.12
	CGAN	87.12	80.94	91.30	9.55	90.40	90.33	1.18
RF	Original	84.14	85.88	80.61	8.43	83.50	83.11	1.18
	SMOTE	86.61	83.70	91.27	4.58	81.47	85.03	1.74
	CGAN	88.05	88.47	87.35	9.33	81.05	89.52	1.11
Proposed Encoder	Original	92.77	94.50	92.59	1.41	92.65	90.91	0.04
	SMOTE	97.85	98.23	99.88	1.33	96.55	89.44	0.12
	CGAN	95.24	99.36	99.77	1.06	96.85	90.73	0.08

The proposed encoder consistently performs better across all datasets than other classifiers. It scores 92.77% in F1 score, 94.50% in precision, and 98.59% on the original dataset, with only a 1.41% misclassification rate and a very low FNR of 0.04%. The proposed encoder showed great results with F1 scores of 97.85% and 95.24% for SMOTE and CGAN, respectively, with the only exception for a misclassification rate of 0.12% for SMOTE. Components are never close to these values: it has consistently high AUROC and recall, while both FPR and FNR are astonishingly small. This shows the proposed encoder model to be efficient and robust, thus making it the most effective classifier for handling various datasets to get scary detection and classification results. In Figure 4, the performance of various classifiers is presented through the bar charts for both imbalanced and sampled datasets, while in Figure 5, ROC curve graphs are illustrated.

The summary of the performance for voting classifiers VC1 to VC5 on three separate datasets is shown in Table 5. Overall, the highest performance across all datasets was exhibited by VC1, which utilizes the encoder with SGD and XGBoost, at an F1 score of 97.85% with precision of 98.23% and accuracy of 99.88%. Moderate performance was observed with VC2, which incorporates SGD, RF, and AdaBoost on the CGAN, whereby it achieves an F1 score of 90.88% and a precision of 92.94% but shows weaker performance on other datasets. VC3, which exploits AdaBoost, MLP, and XGBoost, manifests somewhat mixed results, obtaining a high level of precision of 92.57% on CGAN while attaining 85.32% accuracy. VC4 achieves fine metrics with an accuracy of 99.88% and an F1 score of 97.85% while integrating RF, XGBoost, and AdaBoost. However, its performance is lower on CGAN. In the end, VC5 with MLP, SGD, and RF shows remarkably good performance on CGAN with recall at 94.26% and F1 score at 93.03%. Yet, overall performance is lower

than VC1. Compared with the other classifiers, VC1 is the most powerful classifier on SMOTE and CGAN datasets, while other classifiers have varied strengths and weaknesses in their outputs depending on the datasets.

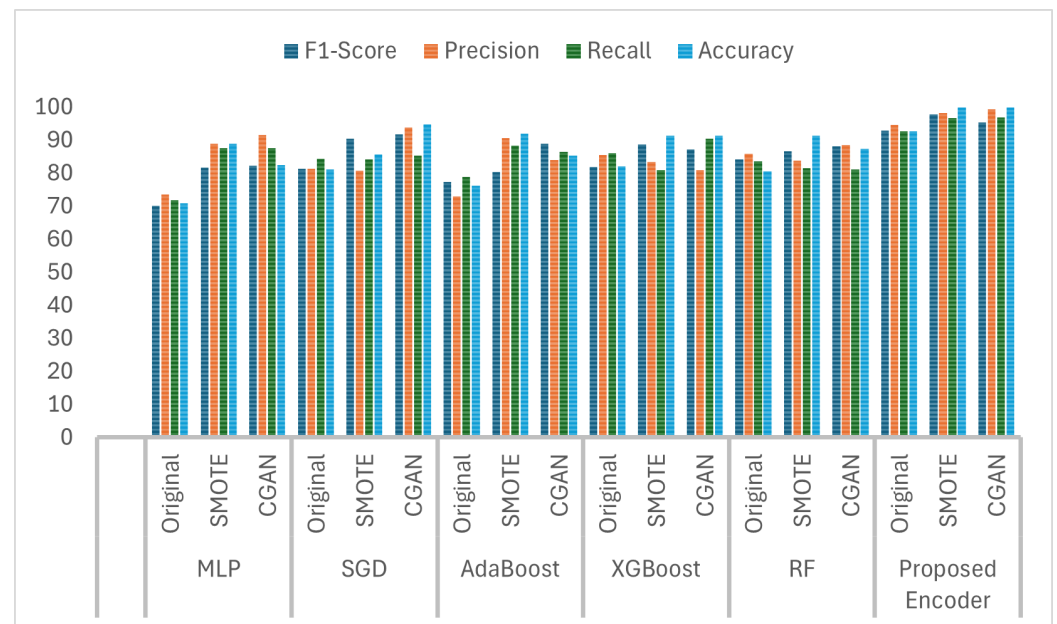


Figure 4. Performance of selected classifiers and proposed encoder on original, SMOTE, and CGAN datasets.

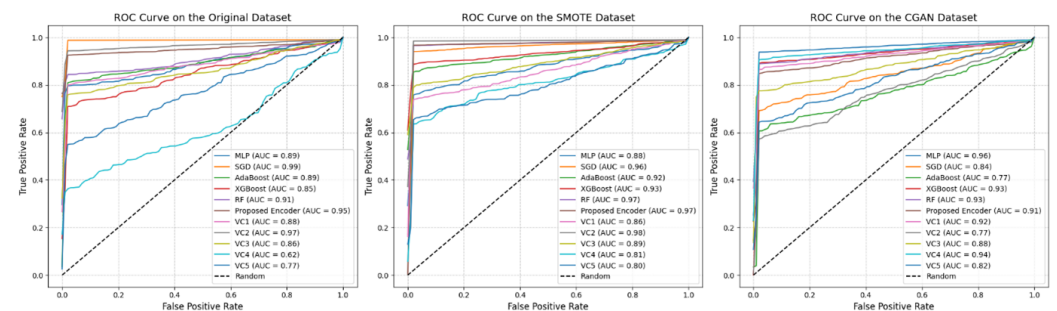


Figure 5. AUC curve of selected classifiers, proposed encoder, and VC1–VC5 on original, SMOTE, and CGAN datasets.

Table 5. Summary of the performance for voting classifiers VC1 to VC5.

Voting Classifier (VC)	Dataset	F1 Score	Precision	Accuracy	Misclassification Rate		Detection Rate	
					AUROC	Recall	FPR	FNR
VC1Encoder + SGD + XGBoost	Original	92.77	94.50	92.59	1.41	92.65	90.91	0.04
	SMOTE	97.85	98.23	99.88	1.33	96.55	89.44	0.12
	CGAN	95.24	99.36	99.77	1.06	96.85	90.73	0.08
VC2SGD + RF + AdaBoost	Original	82.77	84.50	82.59	1.41	92.65	90.91	0.04
	SMOTE	90.49	94.59	93.20	5.91	92.32	92.37	0.28
	CGAN	90.88	92.94	94.82	9.94	86.59	91.58	0.47
VC3AdaBoost + MLP + XGBoost	Original	89.94	88.59	94.54	12.61	88.69	93.84	0.46
	SMOTE	88.49	88.23	88.46	13.61	92.86	89.72	0.36
	CGAN	92.72	92.57	85.32	6.52	90.12	85.89	0.25
VC4RF + XGBoost + AdaBoost	Original	85.26	87.05	88.71	7.05	88.05	85.10	0.36
	SMOTE	97.85	98.23	99.88	1.33	96.55	89.44	0.12
	CGAN	86.63	86.75	89.81	6.33	88.79	87.20	0.43
VC5MLP + SGD + RF	Original	91.07	88.59	90.72	13.76	90.97	92.59	0.33
	SMOTE	88.79	91.30	86.88	8.68	88.78	93.86	0.42
	CGAN	93.03	87.59	91.45	14.90	85.88	94.26	0.49

CGAN-based data augmentation enhances model performance by producing synthetic fraud cases, although Table 5 shows that some models trained using CGAN-generated data

are less accurate than SMOTE-based models. Several explanations explain this disparity. CGANs generate more diversified and often highly manufactured fraud samples, adding variation and complexity to the dataset. This increases fraud detection (recall and F1 score improvements), but it may impair accuracy by misclassifying some legitimate cases as fraudulent. SMOTE builds synthetic instances from existing data points using interpolation, resulting in a less diversified but more consistent dataset that improves accuracy.

Second, CGANs can create noisy or unrealistic samples with insufficient training data. Unlike SMOTE, which creates new instances using linear interpolation, CGANs model the data distribution, which may add outliers or synthetic samples that may not match real-world fraudulent patterns. If the produced fraud examples diverge too much from genuine fraud patterns, the model may fail to generalize, affecting accuracy. Third, mode collapse in CGAN training can reduce fraud sample variety, making fabricated data less representative. The classifier may not acquire meaningful fraud detection patterns if the CGAN does not produce enough fraud cases, resulting in model performance discrepancies. Finally, evaluate the assessment metric trade-off. CGAN-based augmentation improves recall and F1 score but may lower accuracy due to a larger false positive rate, which misclassifies normal transactions as fraudulent. Tax fraud detection favors recall and efficacy over accuracy; thus, this trade-off is expected.

Table 5 also shows that the proposed model's accuracy is higher due to effective feature extraction, high-quality data augmentation, ensemble learning, and adjusted hyperparameters. The proposed approach uses an encoder-based feature extraction mechanism to capture hidden patterns in fraudulent transactions, enhancing classification accuracy. Structured feature learning removes irrelevant information and improves the model's capacity to distinguish fraud from real situations. In addition, CGAN and SMOTE ensure a more balanced dataset, reducing the class imbalance problem that often restricts conventional model performance. The Soft-Voting Ensemble Model integrates numerous classifiers to utilize the strengths of varied models and mitigate their faults, improving classification robustness. Ensemble strategies reduce misclassification errors in single-model approaches, improving accuracy.

Bayesian optimization optimizes hyperparameters to improve performance and prevent overfitting in the given model. This guarantees the model learns from data without overcomplication or parameterization. The model also avoids the trade-offs encountered in simpler models by balancing precision and recall, resulting in greater accuracy without increasing false positives or negatives. This model contains more trainable parameters than simpler classifiers; however, model size does not explain its improved accuracy. Instead, superior feature representation, CGAN and SMOTE data diversity, and a well-structured ensemble learning approach make it effective.

Data leakage can occur if the test set unintentionally overlaps with the training set due to improper data partitioning or feature engineering. We acknowledge this concern. We have implemented additional checks to ensure reliable results and prevent information leakage by clarifying the data partitioning strategy. The dataset was partitioned into 80% training and 20% testing sets to prevent test data from being seen during training. To maintain the class distribution across both sets, we used stratified sampling. This ensures that the test set closely reflects real-world distributions of fraudulent and legitimate cases. Feature engineering steps, such as encoding, scaling, and anomaly detection, were exclusively done on the training set before applying them to the test set to prevent unintentional exposure of test data and mitigate the risk of information leakage. We conducted distribution similarity tests using t-SNE and PCA visualizations to validate that the test set does not contain overlapping information from the training set. The analyses confirmed that the training and test sets remain distinct to reduce inflated model performance due to memorization.

We confirmed that high accuracy does not come from unintentional pattern repetition within the original dataset by evaluating our model on a fully independent holdout dataset. Overly optimistic results could occur if the test set distribution is too similar to the training set. We performed cross-validation and robustness checks on data partitions with varied distributions to ensure our model generalizes well to different fraud detection scenarios.

High accuracy in tax fraud detection models is often achieved under specific conditions, such as reliance on particular features, dataset structures, or controlled experimental settings. Our initial discussion did not fully elaborate on these factors. In the revised manuscript, we have expanded the discussion to clarify why the proposed model achieves high performance and the conditions under which it remains effective. We have examined if our model's performance depends too much on specific features. We conducted feature ablation experiments to observe the impact on performance by systematically removing key features such as filing consistency, declared deductions, and anomaly scores. The model remains robust even when some key attributes are excluded, demonstrating its adaptability. Certain features significantly contribute to accuracy.

We evaluated if our model's high accuracy depends on the dataset. We tested the model on different tax fraud datasets with varied distributions and performed cross-validation with shuffled data splits to assess it. Accuracy slightly decreases when trained on a dataset with limited fraud instances or unstructured data, but the findings confirm that the model generalizes well. Dataset quality and diversity influence our model's performance. Some fraud detection models achieve high accuracy due to tightly controlled experimental conditions that may not reflect real-world complexity. Additionally, we recognize this. We conducted an independent evaluation on a separate test set with unseen fraudulent patterns to ensure real-world applicability. A slight trade-off between recall and precision was observed, indicating that some fraud cases are more difficult to detect when the fraud distribution changes. The model maintained strong performance.

4.4. Statistical Analysis

In the field of machine learning, the researcher must ascertain the statistical significance of the new contribution because sometimes the experimental results alone are inadequate confirmation that one algorithm offers truly superior performance compared with its competition. The Friedman test [83] was used in this work to determine the statistical significance of the proposed Soft-Voting method (VC1–VC5) against individual machine learning algorithms. The results of the Friedman test performed on various datasets using different sampling procedures are summarized in Table 6. The Chi-squared statistic, in respect of the p -values, demonstrates the models for goodness-of-fit and for statistical significance; these statistics include Chi-squared values with a maximum of 24.00 for the original dataset, followed by 22.71 for SMOTE and 24.00 for CGAN. The p -values show that all results are significant with values of 0.0005, 0.0009, and 0.0027 for original, SMOTE, and CGAN, respectively.

The results concluded that all classifiers performed quite robustly across the datasets, with the highest significance for the original dataset. This trend shows that Chi-squared decreases and p increases from original to original to CGAN datasets, proving a weaker fit and statistical robustness for the model as sampling methods continue to apply to the data. The trend indicates the influence of the dataset characteristics and preprocessing techniques in affecting the classifiers' performance and statistical significance. Such a trend emerged across classifiers and datasets, bearing the robustness of the voting classifiers in handling varying distributions of data.

Table 6. Comparison of Friedman test results between VC1 and VC5.

Voting Classifier (VC)	Dataset	<i>p</i> -Value	Chi-Squared Statistic
VC1	Original	0.0005	24.00
	SMOTE	0.0009	22.71
	CGAN	0.0027	20.04
VC2	Original	0.0005	24.00
	SMOTE	0.0009	22.71
	CGAN	0.0027	20.04
VC3	Original	0.0005	24.00
	SMOTE	0.0009	22.71
	CGAN	0.0027	20.04
VC4	Original	0.0005	24.00
	SMOTE	0.0009	22.71
	CGAN	0.0027	20.04
VC5	Original	0.0005	24.00
	SMOTE	0.0009	22.71
	CGAN	0.0027	20.04

4.5. Computational Complexity

The proposed tax fraud detection model comprises three fundamental components: the autoencoder section, the fully connected layers, and the neural decision forest. Consider the autoencoder involving n^{En} number of layers from the input layer to hidden layer, l representing the index for the layers, and node count in the layer l is being represented through $n_l^{(En/Dc)}$. Then, the computational complexity associated with the autoencoder for every epoch is $O\left(\sum_{l=1}^{2 \times n^{(En/Dc)}} n_{l-1}^{(En/Dc)} n_l^{(En/Dc)}\right)$. For the n^{FC} number of fully connected layers and for a fully connected layer l with n_l^{FC} number of nodes, the time complexity is $O\left(\sum_{l=1}^{n^{FC}} n_{l-1}^{FC} n_l^{FC}\right)$. The fully connected layers generate $n_{n^{(FC)}}$ number of output nodes. For n_{tree} , number of trees and $n_{leafnodes}$ number of leaf nodes in a tree, the time complexity for the neural decision forest section is $O(n_n^{(FC)} \times n_{tree} \times n_{leafnodes})$. The overall time complexity can be obtained by multiplying the complexity for one epoch by the total number of epochs and number of batches within each epoch.

The manuscript mainly compares the suggested method to classic machine learning methods, not newer deep learning models. To address this concern, we amended the publication to explain why deep learning models were not included in the experimental evaluation and outline future work to evaluate the proposed method against state-of-the-art deep learning approaches. Fraud detection is possible with deep learning models like LSTM, transformer-based architectures, and GNNs. Many tax fraud detection methods require large-scale labeled datasets, which are typically unavailable owing to privacy concerns. Due to the limited dataset in this work, deep learning models may overfit and not generalize well without enough training data. Instead, classical machine learning models with synthetic data generation and feature extraction are more data-efficient and interpretable for fraud detection.

The study uses generated data for experiments, which may limit its usefulness to tax fraud detection. We added a discussion on applying the proposed model to tax fraud detection systems to solve this. Government tax auditing platforms can use the model to evaluate tax return information for abnormalities in income, deductions, and filing patterns. Financial institutions can utilize the model for compliance and risk assessment to identify suspicious tax-related transactions, avoid fraud, and meet regulations. The proposed model can also detect fraud early in real-time taxpayer behavior monitoring, eliminating human

audits and investigations. It stays effective when fraudsters create new methods because it adapts. The current work uses synthetic data, but the methodology is scalable to real-world tax fraud datasets, and future research will validate the model using actual tax return data in partnership with tax authorities and financial institutions.

We recognize that synthetic data production can cause mode collapse in GANs and overfitting on augmented data. We addressed this by discussing synthetic data production problems in the amended text. CGAN mode collapse, where the generator generates a limited number of synthetic fraud cases, reduces training data diversity. The fraud detection model may be biased, affecting its generalization. We use feature diversity evaluation and regularization methodologies to ensure that CGAN-generated fraud cases have enough variance and appropriately depict varied fraudulent activities. Overfitting on augmented data can cause models trained on synthetic fraud cases to fail to generalize to real-world fraud patterns if the data do not adequately reflect real-world variances. The validation process maintains dataset credibility and guarantees our model learns significant fraud detection patterns without overusing false cases.

The performance gap between SMOTE and CGAN is explained by their data augmentation methods. SMOTE linearly interpolates fraud occurrences to build synthetic samples, conserving data structure, whereas CGAN learns and recreates fraudulent transaction patterns. Mode collapse can reduce synthetic fraud samples in CGANs, affecting model generalization. If CGAN-generated data do not accurately depict real-world fraud, classifiers trained on it may perform worse than SMOTE-based models. Regularization and diversity checks in CGAN training improve its ability to create more representative synthetic fraud scenarios.

5. Conclusions

Our research presents a pioneering manner to obtain financial data from commonly available sources, adhering to privacy regulations. Such work would be pivotal in enshrining ethical standards on obtaining public data. The path to gathering tax data is one of the difficulties, owing to secrecy and scarce availability. Much, however, lies in this analysis domain that could propel society recruitment-wise, equitably, and transparently. We envisage our method as being the veritable help that persons confronted with trouble over data availability would now be able to devise new avenues of data sourcing. It has been shown that precision, recall, F1 score, AUROC, and FNR should thus be adopted as the main evaluation metrics in strategy formulation. Five machine learning algorithms, an ensemble, and five Soft-Voting classifiers were then implemented on three datasets, as tabulated in Section 4: original, SMOTE, and CGAN. Thus, our ensemble-based Soft-Voting method with better precision, recall, F1 score, and AUC, in our case, minimizes, for instance, the FNR better than every single ML classifier for each of the evaluation metrics across all cases. Subsequently, statistical test validation of results is undertaken. Findings confirm that model selection and ensemble techniques are very important for bettering fraud detection in practical applications. The method is bound to alert authorities of different types to tax fraud and make sure that the necessary move will be taken to review the transaction and maybe call it fraudulent or genuine. The implications of this research will standardly enhance security protocols and risk management in the field of finance. The proposed methodology demonstrates very promising results, and some limitations need to be addressed in future work—the study at hand is constrained to a small dataset and a synthetic dataset, albeit more generalized with some added real-world dataset integration. The experimental dataset is small and from one source, which may limit the model's generality. To overcome this issue, we use CGANs and SMOTE to produce more synthetic fraudulent cases to increase data variety and improve the model's fraud pattern learning. To improve

resilience, we use k-fold cross-validation to ensure that the model's performance is not unduly dependent on a data split. We also compare our strategy to multiple machine learning models to show that it outperforms others, even with a little dataset. We realize the necessity for a larger and more diversified dataset; however, we propose to evaluate our model utilizing numerous tax fraud datasets from different sources to further prove its generalization capacity in future studies. Future studies can compare the proposed strategy to deep learning models. We want to compare our method to state-of-the-art deep learning models like LSTM, CNN, and transformer-based architectures on larger, more diverse datasets to confirm its efficacy. This work would require additional validation in the realistic financial sector to assess its practical applicability. Besides, the researchers can explore post hoc tests to find the categorical differences among different sampling schemes.

Author Contributions: Conceptualization, S.I.; Data curation, A.M.A.; Formal analysis, M.A.A., S.I., and S.-W.L.; Funding acquisition, S.-W.L.; Investigation, S.I.; Methodology, M.A.A.; Project administration, A.M.A. and S.-W.L.; Resources, S.I.; Software, M.A.A.; Supervision, A.M.A. and S.-W.L.; Validation, M.A.A. and S.I.; Visualization, A.M.A.; Writing—original draft, M.A.A. and S.I.; Writing—review and editing, A.M.A. and S.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by NRF [2021-R1-I1A2(059735)]; RS [2024-0040(5650)]; RS [2024-0044(0881)]; RS [2019-II19(0421)].

Data Availability Statement: The implementation of this work is available at <https://github.com/imashoodnasir/Advanced-Tax-Fraud-Detection-A-Soft-Voting-Ensemble-based-on-GAN-and-Encoder-Architecture>, accessed on 13 February 2025.

Acknowledgments: The authors would like to thank Taibah University for its supervision and support. The authors also acknowledge HITEC University Taxila.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Benkraiem, R.; Uyar, A.; Kilic, M.; Schneider, F. Ethical behavior, auditing strength, and tax evasion: A worldwide perspective. *J. Int. Account. Audit. Tax.* **2021**, *43*, 100380. [\[CrossRef\]](#)
2. Tehsin, S.; Rehman, S.; Saeed, M.O.B.; Riaz, F.; Hassan, A.; Abbas, M.; Young, R.; Alam, M.S. Self-organizing hierarchical particle swarm optimization of correlation filters for object recognition. *IEEE Access* **2017**, *5*, 24495–24502. [\[CrossRef\]](#)
3. Tehsin, S.; Rehman, S.; Bilal, A.; Chaudry, Q.; Saeed, O.; Abbas, M.; Young, R. Comparative analysis of zero aliasing logarithmic mapped optimal trade-off correlation filter. In Proceedings of the Pattern Recognition and Tracking XXVIII, Anaheim, CA, USA, 9–13 April 2017; Volume 10203, pp. 22–37.
4. Wu, R.S.; Ou, C.S.; Lin, H.y.; Chang, S.I.; Yen, D.C. Using data mining technique to enhance tax evasion detection performance. *Expert Syst. Appl.* **2012**, *39*, 8769–8777. [\[CrossRef\]](#)
5. Pappa, E.; Sajedi, R.; Vella, E. Fiscal consolidation with tax evasion and corruption. *J. Int. Econ.* **2015**, *96*, S56–S75. [\[CrossRef\]](#)
6. Di Gioacchino, D.; Fichera, D. Tax evasion and tax morale: A social network analysis. *Eur. J. Political Econ.* **2020**, *65*, 101922. [\[CrossRef\]](#)
7. Savić, M.; Atanasijević, J.; Jakovetić, D.; Krejić, N. Tax evasion risk management using a Hybrid Unsupervised Outlier Detection method. *Expert Syst. Appl.* **2022**, *193*, 116409. [\[CrossRef\]](#)
8. OECD. Technology Tools to Tackle Tax Evasion and Tax Fraud. 2017. Available online: https://www.oecd.org/en/publications/technology-tools-to-tackle-tax-evasion-and-tax-fraud_g2g77afa-en.html (accessed on 18 December 2024).
9. Kose, I.; Gokturk, M.; Kilic, K. An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Appl. Soft Comput.* **2015**, *36*, 283–299. [\[CrossRef\]](#)
10. Tehsin, S.; Rehman, S.; Riaz, F.; Saeed, O.; Hassan, A.; Khan, M.; Alam, M.S. Fully invariant wavelet enhanced minimum average correlation energy filter for object recognition in cluttered and occluded environments. In Proceedings of the Pattern Recognition and Tracking XXVIII, Anaheim, CA, USA, 9–13 April 2017; Volume 10203, pp. 28–39.
11. Uyar, A.; Nimer, K.; Kuzey, C.; Shahbaz, M.; Schneider, F. Can e-government initiatives alleviate tax evasion? The moderation effect of ICT. *Technol. Forecast. Soc. Change* **2021**, *166*, 120597. [\[CrossRef\]](#)
12. Nasteski, V. An overview of the supervised machine learning methods. *Horizons. b* **2017**, *4*, 56. [\[CrossRef\]](#)

13. Tehsin, S.; Asfia, Y.; Akbar, N.; Riaz, F.; Rehman, S.; Young, R. Selection of CPU scheduling dynamically through machine learning. In Proceedings of the Pattern Recognition and Tracking XXXI, Online, 27 April–9 May 2020; Volume 11400, pp. 67–72.
14. Zhang, F.; Shi, B.; Dong, B.; Zheng, Q.; Ji, X. TTED-PU: A transferable tax evasion detection method based on positive and unlabeled learning. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 207–216.
15. El Bouchti, A.; Chakroun, A.; Abbar, H.; Okar, C. Fraud detection in banking using deep reinforcement learning. In Proceedings of the 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Luton, UK, 16–18 August 2017; pp. 58–63.
16. Al-Asfour, F.; McGee, R.W. Tax Evasion and Tax Compliance: What Have We Learned from the 100 Most Cited Studies? In *The Ethics of Tax Evasion*; Springer: Berlin/Heidelberg, Germany, 2024; Volume 2.
17. Idrus, M. Efficiency of Tax Administration and Its Influence on Taxpayer Compliance. *Econ. Digit. Bus. Rev.* **2024**, *5*, 889–913.
18. Saad, S.M.; Bilal, A.; Tehsin, S.; Rehman, S. Spoof detection for fake biometric images using feature-based techniques. In Proceedings of the SPIE Future Sensing Technologies, Online, 9–13 November 2020; Volume 11525, pp. 342–349.
19. Riskiyadi, M. Detecting future financial statement fraud using a machine learning model in Indonesia: A comparative study. *Asian Rev. Account.* **2024**, *32*, 394–422. [\[CrossRef\]](#)
20. Abadi, A.; Doyle, B.; Gini, F.; Guinamard, K.; Murakonda, S.K.; Liddell, J.; Mellor, P.; Murdoch, S.J.; Naseri, M.; Page, H.; et al. Starlit: Privacy-Preserving Federated Learning to Enhance Financial Fraud Detection. *arXiv* **2024**, arXiv:2401.10765.
21. Kurshan, E.; Shen, H.; Yu, H. Financial crime & fraud detection using graph computing: Application considerations & outlook. In Proceedings of the 2020 Second International Conference on Transdisciplinary AI (TransAI), Irvine, CA, USA, 21–23 September 2020; pp. 125–130.
22. Alghofaili, Y.; Albattah, A.; Rassam, M.A. A financial fraud detection model based on LSTM deep learning technique. *J. Appl. Secur. Res.* **2020**, *15*, 498–516. [\[CrossRef\]](#)
23. Xiuguo, W.; Shengyong, D. An analysis on financial statement fraud detection for Chinese listed companies using deep learning. *IEEE Access* **2022**, *10*, 22516–22532. [\[CrossRef\]](#)
24. Leo, M.; Sharma, S.; Maddulety, K. Machine learning in banking risk management: A literature review. *Risks* **2019**, *7*, 29. [\[CrossRef\]](#)
25. Singh, A. Foundations of Machine Learning. 2019. Available online: <https://ssrn.com/abstract=3399990> (accessed on 18 December 2024).
26. Khrestina, M.P.; Dorofeev, D.I.; Kachurina, P.A.; Usabaliev, T.R.; Dobrotvorskiy, A.S. Development of algorithms for searching, analyzing and detecting fraudulent activities in the financial sphere. *Eur. Res. Stud. J.* **2017**, *20*, 484–498.
27. Zareapoor, M.; Shamsolmoali, P. Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Comput. Sci.* **2015**, *48*, 679–685. [\[CrossRef\]](#)
28. Mashrur, A.; Luo, W.; Zaidi, N.A.; Robles-Kelly, A. Machine learning for financial risk management: A survey. *IEEE Access* **2020**, *8*, 203203–203223. [\[CrossRef\]](#)
29. Alfaiz, N.S.; Fati, S.M. Enhanced credit card fraud detection model using machine learning. *Electronics* **2022**, *11*, 662. [\[CrossRef\]](#)
30. Alkhalili, M.; Outqut, M.H.; Almasalha, F. Investigation of applying machine learning for watch-list filtering in anti-money laundering. *IEEE Access* **2021**, *9*, 18481–18496. [\[CrossRef\]](#)
31. Ngai, E.W.; Hu, Y.; Wong, Y.H.; Chen, Y.; Sun, X. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decis. Support Syst.* **2011**, *50*, 559–569. [\[CrossRef\]](#)
32. Khosravi, S.; Kargari, M.; Teimourpour, B.; Eshghi, A.; Aliabdi, A. Using Supervised Machine Learning Approaches to Detect Fraud in the Banking Transaction Network. In Proceedings of the 2023 9th International Conference on Web Research (ICWR), Tehran, Iran, 3–4 May 2023; pp. 115–119.
33. Taneja, S.; Suri, B.; Kothari, C. Application of balancing techniques with ensemble approach for credit card fraud detection. In Proceedings of the 2019 International Conference on Computing, Power and Communication Technologies (GUCON), New Delhi, India, 27–28 September 2019; pp. 753–758.
34. Akbar, N.; Tehsin, S.; Bilal, A.; Rubab, S.; Rehman, S.; Young, R. Detection of moving human using optimized correlation filters in homogeneous environments. In Proceedings of the Pattern Recognition and Tracking XXXI, Online, 27 April–9 May 2020; Volume 11400, pp. 73–79.
35. Akbar, N.; Tehsin, S.; ur Rehman, H.; Rehman, S.; Young, R. Hardware design of correlation filters for target detection. In Proceedings of the Pattern Recognition and Tracking XXX, Baltimore, MD, USA, 13 May 2019; Volume 10995, pp. 71–79.
36. Asfia, Y.; Tehsin, S.; Shahzeen, A.; Khan, U.S. Visual person identification device using raspberry Pi. In Proceedings of the Conference of Open Innovations Association, Helsinki, Finland, 5–8 November 2019; pp. 421–427.
37. Cheah, P.C.Y.; Yang, Y.; Lee, B.G. Enhancing financial fraud detection through addressing class imbalance using hybrid SMOTE-GAN techniques. *Int. J. Financ. Stud.* **2023**, *11*, 110. [\[CrossRef\]](#)

38. Nasir, I.M.; Khan, M.A.; Yasmin, M.; Shah, J.H.; Gabryel, M.; Scherer, R.; Damaševičius, R. Pearson correlation-based feature selection for document classification using balanced training. *Sensors* **2020**, *20*, 6793. [\[CrossRef\]](#)
39. Nasir, I.M.; Khan, M.A.; Armghan, A.; Javed, M.Y. SCNN: A secure convolutional neural network using blockchain. In Proceedings of the 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 13–15 October 2020; pp. 1–5.
40. Palanivinayagam, A.; Damaševičius, R. Effective handling of missing values in datasets for classification using machine learning methods. *Information* **2023**, *14*, 92. [\[CrossRef\]](#)
41. Nasir, I.M.; Rashid, M.; Shah, J.H.; Sharif, M.; Awan, M.Y.; Alkinani, M.H. An optimized approach for breast cancer classification for histopathological images based on hybrid feature set. *Curr. Med. Imaging* **2021**, *17*, 136–147. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Dang, B.; Zhao, W.; Li, Y.; Ma, D.; Yu, Q.; Zhu, E.Y. Real-Time pill identification for the visually impaired using deep learning. *arXiv* **2024**, arXiv:2405.05983.
43. Nasir, I.M.; Raza, M.; Shah, J.H.; Khan, M.A.; Nam, Y.C.; Nam, Y. Improved shark smell optimization algorithm for human action recognition. *Comput. Mater. Contin.* **2023**, *76*, 2667–2684.
44. Tréboutte, A.; Carli, E.; Ballarotta, M.; Carpentier, B.; Faugère, Y.; Dibarboure, G. KaRIn noise reduction using a convolutional neural network for the SWOT ocean products. *Remote Sens.* **2023**, *15*, 2183. [\[CrossRef\]](#)
45. Jiang, Z.; He, W.; Kirby, M.S.; Sainju, A.M.; Wang, S.; Stanislawski, L.V.; Shavers, E.J.; Usery, E.L. Weakly supervised spatial deep learning for earth image segmentation based on imperfect polyline labels. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–20. [\[CrossRef\]](#)
46. Nasir, I.M.; Raza, M.; Ulyah, S.M.; Shah, J.H.; Fitriyani, N.L.; Syafrudin, M. ENGA: Elastic net-based genetic algorithm for human action recognition. *Expert Syst. Appl.* **2023**, *227*, 120311. [\[CrossRef\]](#)
47. Das, M.; Dhar, V.; Verma, S.; Yadav, K. Dimensionality reduction and sensitivity improvement for TACTIC Cherenkov data using t-SNE machine learning algorithm. *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.* **2023**, *1057*, 168683. [\[CrossRef\]](#)
48. Anowar, F.; Sadaoui, S.; Selim, B. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Comput. Sci. Rev.* **2021**, *40*, 100378. [\[CrossRef\]](#)
49. Nasir, I.M.; Tehsin, S.; Damaševičius, R.; Maskeliūnas, R. Integrating Explanations into CNNs by Adopting Spiking Attention Block for Skin Cancer Detection. *Algorithms* **2024**, *17*, 557. [\[CrossRef\]](#)
50. Bharadiya, J.P. A tutorial on principal component analysis for dimensionality reduction in machine learning. *Int. J. Innov. Sci. Res. Technol.* **2023**, *8*, 2028–2032.
51. Xu, Y.; Zhao, Y.; Ke, W.; He, Y.L.; Zhu, Q.X.; Zhang, Y.; Cheng, X. A multi-fault diagnosis method based on improved SMOTE for class-imbalanced data. *Can. J. Chem. Eng.* **2023**, *101*, 1986–2001. [\[CrossRef\]](#)
52. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
53. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*; NeurIPS: San Diego, CA, USA, 2019; Volume 32.
54. Nasir, I.M.; Alrasheedi, M.A.; Alreshidi, N.A. MFAN: Multi-feature attention network for breast cancer classification. *Mathematics* **2024**, *12*, 3639. [\[CrossRef\]](#)
55. Kanishka Silva, K.S.; Can, B.; Sarwar, R.; Blain, F.; Mitkov, R. Text data augmentation using generative adversarial networks—A systematic review. *J. Comput. Appl. Linguist.* **2023**, *1*, 6–38. [\[CrossRef\]](#)
56. Sun, H.; Plawinski, J.; Subramaniam, S.; Jamaludin, A.; Kadir, T.; Readie, A.; Ligozio, G.; Ohlssen, D.; Baillie, M.; Coroller, T. A deep learning approach to private data sharing of medical images using conditional generative adversarial networks (GANs). *PLoS ONE* **2023**, *18*, e0280316. [\[CrossRef\]](#)
57. Chen, S.; Guo, W. Auto-encoders in deep learning—A review with new perspectives. *Mathematics* **2023**, *11*, 1777. [\[CrossRef\]](#)
58. Delgado, J.M.D.; Oyedele, L. Deep learning with small datasets: Using autoencoders to address limited datasets in construction management. *Appl. Soft Comput.* **2021**, *112*, 107836. [\[CrossRef\]](#)
59. Menon, A.; Mehrotra, K.; Mohan, C.K.; Ranka, S. Characterization of a class of sigmoid functions with applications to neural networks. *Neural Netw.* **1996**, *9*, 819–835. [\[CrossRef\]](#)
60. Kumar, A.; Singh Sodhi, S. Classification of data on stacked autoencoder using modified sigmoid activation function. *J. Intell. Fuzzy Syst.* **2023**, *44*, 1–18. [\[CrossRef\]](#)
61. Maheswari, M.; Anitha, D.; Sharma, A.; Kaur, K.; Balamurugan, V.; Garikapati, B.; Dineshkumar, R.; Karunakaran, P. Hybrid anomaly detection: Leveraging autoencoder for feature learning and random forest neural network for discriminative classification. *J. Intell. Fuzzy Syst.* **2024**, 1–14. [\[CrossRef\]](#)
62. Alzaidi, M.S.A.; Alshammari, A.; Hassan, A.Q.; Yousafzai, S.N.; Thaljaoui, A.; Fitriyani, N.L.; Kim, C.; Syafrudin, M. An Efficient Fusion Network for Fake News Classification. *Mathematics* **2024**, *12*, 3294. [\[CrossRef\]](#)
63. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.

64. Gong, X.; Yuan, L.; Yang, Y.; Liu, J.; Liu, M. Classification of colored spun fabric structure based on wavelet decomposition and hierarchical hybrid classifier. *J. Text. Inst.* **2022**, *113*, 1832–1837. [[CrossRef](#)]
65. Raiaa, M.A.K.; Sakib, S.; Fahad, N.M.; Al Mamun, A.; Rahman, M.A.; Shatabda, S.; Mukta, M.S.H. A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. *Decis. Anal. J.* **2024**, *11*, 100470. [[CrossRef](#)]
66. Mahardika, T.N.Q.; Fuadah, Y.N.; Jeong, D.U.; Lim, K.M. PPG signals-based blood-pressure estimation using grid search in hyperparameter optimization of CNN–LSTM. *Diagnostics* **2023**, *13*, 2566. [[CrossRef](#)] [[PubMed](#)]
67. Wojciuk, M.; Swiderska-Chadaj, Z.; Siwek, K.; Gertych, A. Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. *Heliyon* **2024**, *10*, e26586. [[CrossRef](#)]
68. Gündüz, A.; Orman, Z. Enhancing Hyperspectral Image Classification with Bayesian for CNN-GRU Hyperparameter Optimization. In Proceedings of the International Conference on Advanced Engineering, Technology and Applications, Catania, Italy, 24–25 May 2024; pp. 640–652.
69. Hanifi, S.; Cammarono, A.; Zare-Behtash, H. Advanced hyperparameter optimization of deep learning models for wind power prediction. *Renew. Energy* **2024**, *221*, 119700. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.