iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71636

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

AI-Based Tax Compliance Monitoring and Fraud Detection for Taxpaying Citize

Manjunath Narayana Mavalangi[1], Madhusudhan M V[2]
Department of CSE/ Presidency University, Bangalore, India

Abstract: ***The increasing complexity of financial transactions and tax evasion strategies has necessitated the development of intelligent, data-driven systems to detect fraudulent activity and ensure compliance. This study introduces a robust hybrid framework that leverages Sparse Autoencoders for feature extraction, Neural Decision Forests for high-accuracy classification, and statistical feature engineering for behavioral analysis. The system is trained on a curated dataset of approximately 10,000 financial transactions, encompassing features such as transaction type, account balances, timing, and recipient risk scores, with derived metrics like transaction velocity, deviation, and balance changes. The architecture is designed to identify anomalous patterns, assess evasion probabilities, and flag high-risk accounts. Advanced machine learning models are employed to address challenges such as class imbalance, dynamic user behavior, and hidden fraud patterns. Compared to traditional rule-based methods, the proposed framework enhances fraud detection accuracy while minimizing false positives, offering a scalable solution for regulatory bodies to audit financial flows and improve transparency in tax collection systems.***

Keywords: Financial Fraud Detection, Tax Evasion, Sparse Autoencoder, Neural Decision Forest, Transaction Analysis, Anomaly Detection, Risk Assessment, Machine Learning.

I. INTRODUCTION

Tax fraud, characterized by the intentional manipulation or misrepresentation of financial information to evade tax liabilities, poses a serious threat to the economic stability of governments. Globally, this challenge results in billions of dollars in lost revenue each year. Traditional fraud detection approaches, such as manual audits and rule-based expert systems, have served as the backbone of many tax administration strategies. However, these methods suffer from critical limitations: they are resource-intensive, dependent on domain expertise, and often unable to adapt to the dynamic nature of fraud schemes. These approaches also exhibit high false positive rates due to their rigid and static rule formulations [1].

With the proliferation of digital financial systems and the availability of large-scale transactional data, Machine Learning (ML) has emerged as a promising tool in the domain of tax fraud detection. ML algorithms are capable of identifying complex, non-linear relationships within high-dimensional datasets, enabling the discovery of fraudulent behaviors that are not easily detectable through conventional techniques. These models can be trained to recognize key fraud indicators such as irregular transaction volumes, sudden changes in account balances, abnormal transaction velocity, and suspicious temporal patterns [2].

In this context, Artificial Neural Networks (ANNs) have demonstrated strong performance in learning fraudulent behavior from historical tax data. Recent studies have shown that ANN-based models can achieve accuracy levels above 90%, with high recall and area-under-the-curve (AUC) metrics when applied to income tax datasets [3]. These models not only capture the nonlinear dependencies between input variables but are also effective in highlighting features such as the frequency and amount of transactions, origin-destination relationships, deviation from typical taxpayer behavior, and the risk scores associated with recipients or businesses.

However, relying solely on supervised learning methods, such as ANNs, can be limiting due to their dependence on labeled audit data—which often covers only a fraction of real-world cases. In contrast, unsupervised models can analyze all available data but may lack the specificity needed for accurate fraud detection. A more effective solution lies in hybrid systems that leverage the strengths of both paradigms, allowing for generalized learning from all transaction data while maintaining high classification accuracy on labeled cases [4].

To overcome these limitations, this study proposes a hybrid framework combining Sparse Autoencoders for unsupervised feature extraction, Time Series Forests to analyze sequential transaction behavior over time, and Neural Decision Forests to perform the final classification. Sparse Autoencoders reduce the dimensionality of the input data while preserving fraud-relevant patterns. Time Series Forests help detect anomalies based on temporal features such as transaction hour, time since last transaction, and transaction velocity.

Neural Decision Forests enhance interpretability and robustness by integrating decision-tree logic with deep neural representations .
In addition to the core transactional features, this study also evaluates fraud prediction based on contextual attributes including account origin and destination behavior, historical balance patterns, taxpayer risk scores, and deviation from individual financial norms. By modeling these multidimensional inputs through a unified hybrid learning framework, the proposed system aims to deliver scalable and highly accurate fraud detection—ultimately supporting tax agencies in reducing audit costs, improving compliance, and recovering lost revenue.

II. LITERATURE REVIEW

Tax fraud detection and compliance monitoring have seen significant advancements with the application of machine learning and deep learning approaches. Alexopoulos et al. [5-6] proposed a network-based approach utilizing the VAT transaction network's Laplacian matrix combined with scalable machine learning algorithms, achieving detection of approximately 50% of VAT fraud cases. This method effectively leveraged the complex network structure inherent in VAT transactions to identify anomalies. Murorunkwere et al. [7] evaluated multiple supervised machine learning models, including Artificial Neural Networks (ANN), Logistic Regression, Decision Trees, Random Forests, GaussianNB, and XGBoost, to predict tax fraud. The study found that ANN outperformed other models, effectively identifying key fraud indicators such as business age, domestic operations, import/export activities, absence of reported losses, geographic location, and specific tax registrations.
Tax et al. [8] outlined a research agenda for applying machine learning in e-commerce fraud detection, discussing organizational challenges and proposing future research directions. Ngai et al. [9] conducted a comprehensive review of data mining techniques used for financial fraud detection, emphasizing their effectiveness in detecting fraudulent activities. Phua et al. [10] provided a survey of data mining approaches for fraud detection, concluding that combining multiple techniques improves detection rates.
Kirkos et al. [11] applied decision trees, neural networks, and Bayesian belief networks to detect fraudulent financial statements, achieving high accuracy. Perols [12] compared logistic regression, decision trees, and neural networks for fraud detection, finding logistic regression to be competitive with more complex models. Fanning and Cogger [13] utilized neural networks to detect fraud in financial statements, showing promising results.
Kou et al. [14] reviewed fraud detection techniques in banking and finance, highlighting the role of data mining and machine learning. Bhattacharyya et al. [15] explored data mining techniques for credit card fraud detection, tackling challenges like imbalanced datasets and feature selection. Bolton and Hand [16] discussed statistical methods for fraud detection, emphasizing the role of unsupervised learning techniques in identifying anomalies.
Beneish [17] developed a model using financial ratios to detect earnings manipulation, which provided a foundation for modern fraud detection methods. Chen et al. [18] applied machine learning techniques to detect fraudulent financial reporting, demonstrating the effectiveness of support vector machines (SVM). Lin et al. [19] proposed a framework combining data mining and forensic accounting techniques to detect financial statement fraud.
Yue et al. [20] developed a hybrid model integrating clustering and classification techniques for fraud detection in telecommunications. Sánchez et al. [21] used evolutionary algorithms to optimize fraud detection models, improving accuracy while reducing false positives. Van Vlasselaer et al. [22] introduced a network-based approach to detect VAT carousel fraud, leveraging relationships between entities to identify suspicious activities.
Jans et al. [23] applied process mining techniques to internal auditing, detecting anomalies in business processes that indicated fraud. Hoogs et al. [24] developed a genetic algorithm-based approach for detecting fraud in financial transactions. Vatsa et al. [25] proposed a game-theoretic approach to credit card fraud detection, modeling interactions between fraudsters and detection systems.

III. PROPOSED METHODOLOGY

The Tax-AI methodology integrates Principal Component Analysis (PCA) for feature selection, Autoencoders for anomaly detection, and Isolation Forest for fraud classification. This combination allows the model to efficiently analyze financial transaction patterns and detect fraudulent activities. It ensures high accuracy and robust performance, especially when dealing with imbalanced datasets where fraudulent cases are rare. The hybrid approach enhances scalability, making it suitable for analyzing large-scale financial datasets.

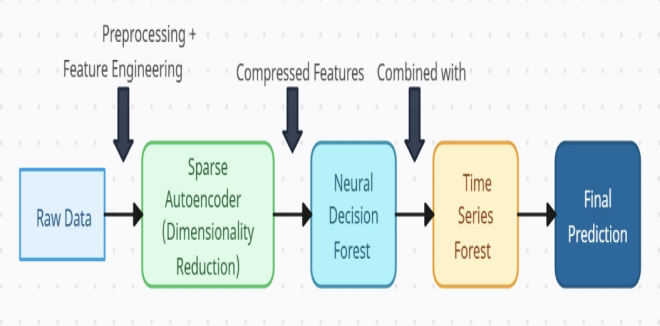


Figure 1:System Architecture for Fraud and Tax Evasion Detection

1) Data Collection:The first stage of the architecture is Raw Data Collection, which involves gathering comprehensive input from various sources relevant to taxation. This includes data from individual tax filings, transactional histories, and administrative records maintained by governmental tax departments. The dataset encompasses multiple types of information: numerical data such as reported income, deductions, and tax credits; categorical data like filing status, employment category, or industry type; and temporal data reflecting patterns over time—such as filing dates, payment schedules, or delays. The objective of this stage is to compile a rich and diverse dataset that accurately represents taxpayer behavior, laying a solid foundation for identifying patterns that may indicate fraudulent activity. High-quality raw data is crucial, as the effectiveness of all subsequent machine learning processes depends on the reliability and depth of this input.

2) Preprocessing:Once the dataset is collected, it undergoes preprocessing to enhance data quality, remove inconsistencies, and ensure compatibility with machine learning models. The preprocessing steps include:

- Data Cleaning: Removing duplicate, inconsistent, or missing values.
- Normalization:Scaling numerical features to a common range using min-max normalization

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X′is the normalized value, X is theoriginal value, and Xmin,Xmax are the minimum and maximum values in the dataset.

- Handling Missing Data: Using interpolation or imputation techniques to fill missing values based on statistical methods such as mean, median, or K-nearest neighbor imputation.
- Encoding Categorical Features: Converting categorical attributes into numerical representations using one-hot encoding or label encoding for better model compatibility.

  In the context of a tax fraud detection system, Preprocessing and Feature Engineering represent foundational steps to ensure the robustness and reliability of downstream machine learning algorithms. From a technical perspective, raw data obtained from tax records and financial transactions often contain inconsistencies, missing entries, outliers, or noise, which can adversely affect model performance. Hence, data cleaning is first employed to eliminate duplicate rows, handle missing values using imputation techniques (e.g., mean, median substitution or KNN imputation), and remove or cap outliers using statistical methods like z-score or IQR filtering.

3) Sparse AutoEncoder

The Sparse Autoencoder (SAE) is a type of unsupervised neural network designed specifically for learning compressed, high-level feature representations from input data. It plays a pivotal role in the architecture of tax fraud detection systems by enabling dimensionality reduction while preserving critical information patterns relevant to identifying anomalous or fraudulent behavior.

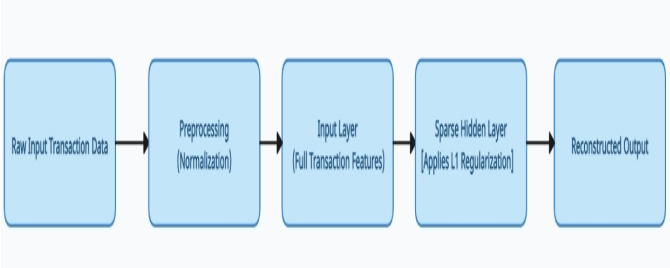


Figure 2:Sparse Autoencoder Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

The Sparse Autoencoder (SAE) plays a crucial role in tax fraud detection systems by enabling the extraction of compact, meaningful representations from high-dimensional financial data. As an unsupervised neural network, the SAE is composed of two primary components: an encoder and a decoder. The encoder transforms the original input vector xxx into a lower-dimensional latent space representationz using the transformation $z = \sigma(Wx + b)$, where W and b represent the learnable weights and biases, and σ is a non-linear activation function such as ReLU or sigmoid. This encoding process reduces the dimensionality of the input data while preserving the most relevant structural information required for downstream tasks like fraud classification.

a) Encoder Function (Dimensionality Reduction):

The encoder is the first component of the autoencoder architecture. It transforms the high-dimensional input vector$x \in R^n$ into a lower-dimensional latent representation$z \in R^m$, where $m < n$.

The transformation is given by:

$$z = f_{encoder(X)=\sigma(Wx+b)}$$

- W: Weight matrix between input and hidden layers.
- b: Bias vector.
- σ: Activation function (commonly ReLU or sigmoid).
- z: Encoded representation (compressed features).

b) Sparsity Constraint (Feature Selection):

To force the model to learn only the most informative features, a sparsity constraint is applied. This constraint ensures that only a small number of neurons in the hidden layer are active (non-zero) for any given input.

This is done by adding a Kullback–Leibler (KL) divergence term to the loss function:

$$KL(\rho||\rho i) = \rho log(\frac{\rho}{\rho_i}) + (1-\rho)log(\frac{1-\rho}{1-\rho i})$$

- ρ: Desired average activation of hidden units (e.g., 0.05).
- ρi: Empirical average activation of hidden unit iii
- Decoder Function (Reconstruction):

The decoder function is a key component of an autoencoder (a type of neural network used for unsupervised learning and feature compression). Its main job is to reconstruct the original input from a compressed (lower-dimensional) representation, called the latent vector.

The decoder function reconstructs the original input x^ from the latent representation z using the equation:

$$\hat{x} = f_{decoder}(Z) = \sigma(W'z + b')$$

Here,W'and b′are the decoder's weights and biases, and σis the activation function applied to recover the input structure from compressed features.

4) Sparse HiddenLayer (Applies L1 Regularization)

This is the core of the Sparse Autoencoder. It applies a sparsity constraint using L1 regularization, encouraging only a small number of neurons to be active at any time. The compression process reduces the feature space while retaining only the most salient patterns.

- Activation Function: Often a nonlinear function like ReLU or sigmoid.
- Sparsity Enforcement:

$$L = ||x - \hat{x}||^2 + \lambda \sum_i |hi|$$

Where hiare the activations in the hidden layer, and λ controls the sparsity level. This forces the network to ignore irrelevant data, promoting interpretability and generalization.

5) *Reconstructed Output*

The decoder attempts to reconstruct the original input from the compressed latent representation. The network is trained to minimize reconstruction loss (typically mean squared error):

$$L = ||x - \hat{x}||^2$$

A low reconstruction error indicates that the hidden layer has effectively captured the essential structure of the input data.

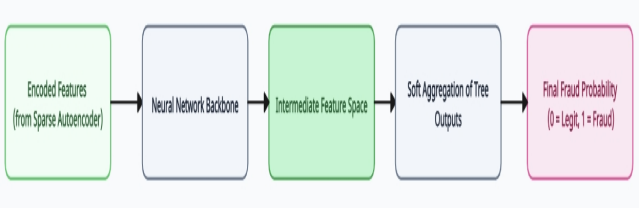


Figure 3:Neural Decision Forest Architecture

This architecture combines the feature extraction capabilities of neural networks with the interpretability and structured decision-making of decision forests to predict fraud probabilities in a probabilistic and explainable manner.

a) Encoded Features (Input Layer)

The input to the NDF is a compressed feature vector z, produced by a Sparse Autoencoder. This vector retains the essential structure of the data while reducing noise and dimensionality**.**

Step 2: Neural Network Backbone

The encoded vector z is passed through **fully connected layers** (also called dense layers) to map it into a more informative latent feature space.

Each hidden layer applies a transformation:

$$h_l = \sigma(W_l h_{l-1} + b_l$$

where:

$W_l$and b_l are weights and biases of layer l,

σ is a non-linear activation function like ReLU or sigmoid,

h0=zis the input from the autoencoder

Step 3: Intermediate Feature Space

The output of the final fully connected layer is the **intermediate feature representation**, which captures the refined characteristics of the input data suitable for tree-based decision making.

Step 4: Soft Decision Trees (Probabilistic Trees)

Unlike classic hard-threshold trees, **soft decision trees** allow differentiable routing of samples using **sigmoid functions** at each split node:

$$p_{left} = \sigma(\theta^T h) \quad and\ p_{right} = 1 - p_{left}$$

Each path in the tree contributes **probabilistically** to the output class, not just a single leaf node.

Step 5: Ensemble Output Aggregation

Each of the T decision trees produces a probability distribution Pt(y□z)over the target classes (0 = Legit, 1 = Fraud).

The final prediction $\hat{y}NDF = \frac{1}{T}\sum_{t=1}^{T} P_t(y\backslash z)$NDF is the average of the soft outputs of all trees:

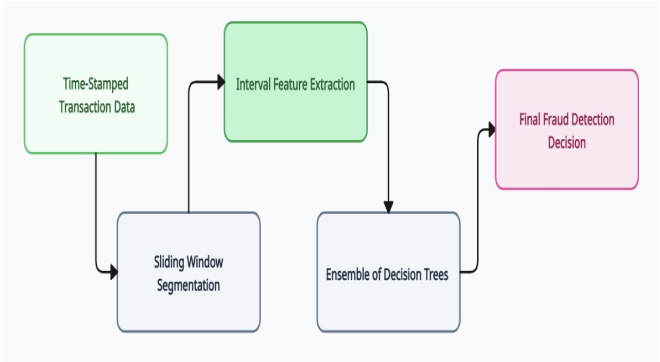


Figure 4: Time Series Forest Architecture

This architecture aims to leverage temporal patterns in financial transaction data to identify potential fraud. By segmenting transaction histories using sliding windows, extracting statistical features from those windows, and applying an ensemble of decision trees, the model makes accurate and interpretable

- Time-Stamped Transaction Data

At the core of this model lies temporal data collected from various transactional activities such as tax filings, online payments, or business records. Each transaction has a timestamp associated with it, making it a time series. The input data is represented as:

$$X = \{(t1, x1), (t2, x2) \ldots\ldots\ldots\ldots (tn, xn)\}$$

Where:

t_i: Timestamp of the i^{it}transaction

x_i: Feature vector for the i^{th}transaction

(e.g., amount, category, location)

This step ensures the system uses historical behavior patterns over time to detect anomalies.

- Sliding Window Segmentation

Time-series data is divided into smaller, manageable segments using a sliding window technique. Each window captures short-term behavior within a fixed time interval:

Window size w defines the number of data points in each segment.

The window "slides" across time with a step size s, generating overlapping segments.

For a time series of length nnn, and a window size w, the number of segments generated is:

$$\text{Segments} = \left[\frac{n - w}{s} + 1\right]$$

This method allows capturing recent patterns that may indicate fraudulent behavior (e.g., sudden spike in claims).

- Interval Feature Extraction

Once the time-stamped transaction data is segmented into fixed-length windows using a sliding window method, the next critical step is to transform these raw temporal segments into a numerical feature space. This transformation allows machine learning models (like decision trees) to process time series data efficiently.

- Mean (Average Value)

Captures the central tendency of values in the window:

$$\mu = \frac{1}{w}\sum_{i=1}^{w} x_i$$

This gives the average value of the feature over the time interval. It helps in identifying elevated transaction amounts or behavior.

- Standard Deviation (Dispersion of Values)

Measures **variability** or **volatility** in the window:

$$\sigma = \sqrt{\frac{1}{w}\sum_{i=1}^{w} (x_i - \mu)^2}$$

High standard deviation may indicate erratic or suspicious behavior.

- Slope / Trend (Linear Regression Coefficient)

Detects the **directional movement** in transaction values (increasing or decreasing pattern):

$$x_i = \alpha + \beta t_i + \varepsilon_i$$

Where:

βis the **slope** of the best-fit line over the time window.

Positive β: upward trend;

Negative β: downward trend.

This helps in identifying sudden spikes or gradual increases in claim..

DATA SET DEATAILS:

The Financial Transaction Fraud Dataset consists of approximately 10,000 structured records, each representing a detailed financial transaction used for fraud detection applications. The dataset includes 17 essential features that capture various behavioral and monetary aspects of transactions, such as Transaction_Type, Amount, Name_Orig, Name_Dest, Old_Balance_Origin, New_Balance_Origin, Old_Balance_Destination, New_Balance_Destination, Account_Number,Cash_In, Cash_Out, Transaction_Velocity, Amount_Deviation,Recipient_Risk_Score,Time_Since_Last_Transaction, Transaction_Hour, and a binary label Fraud indicating whether the transaction is legitimate (0) or fraudulent (1).

Despite natural variations in transaction volume, frequency, and user behavior, the dataset maintains consistency and high quality, supporting robust training of machine learning models. During preprocessing, numerical features are normalized to a common scale, while categorical variables such as transaction type are one-hot encoded to enable compatibility with downstream models. This dataset is well-suited for complex classification tasks and enables the development of advanced fraud detection algorithms, including sparse autoencoders, neural decision forests, and time-aware tree-based models. By capturing intricate transaction patterns and contextual behavior, the dataset facilitates the construction of accurate, automated, and real-time fraud detection systems in financial domains.

	Amount	Name_Orig	Name_Dest	Old_Balance_Origin	New_Balance_Origin
0	231416.415797	C0	M0	990531.516219	759115.100422
1	362062.978088	C1	M1	432370.768138	70082.819369
2	150659.240941	C2	M2	32320.767588	-118338.473354
3	231416.984186	C3	M3	302436.983039	71019.998854
4	109776.944466	C4	M4	650787.732206	541010.787740

	Old_Balance_Destination	New_Balance_Destination	Account_Number	Cash_In
0	558538.117526	789954.533324	ACC0	1
1	571802.152457	933865.130544	ACC1	1
2	692202.333029	842861.573971	ACC2	1
3	588699.826671	820116.810856	ACC3	0
4	340580.913997	450029.572291	ACC4	0

	Cash_Out	...	Transaction_Type_CASH_OUT	Transaction_Type_DEBIT	\
0	0	...	False	False	
1	0	...	False	True	
2	1	...	False	False	
3	1	...	False	True	
4	1	...	False	True	

	Transaction_Type_PAYMENT	Transaction_Type_TRANSFER	Annual_Income	\
0	True	False	2.776997e+06	
1	False	False	4.344756e+06	
2	False	True	1.807911e+06	
3	False	False	2.777004e+06	
4	False	False	1.317323e+06	

	Calculated_Tax	Tax_Gap	Income_Tax_Ratio	Balance_Change_Origin	\
0	5.330991e+05	5.330991e+05	0.191970	-231416.415797	
1	1.003427e+06	1.003427e+06	0.230951	-362287.948769	
2	2.423733e+05	2.423733e+05	0.134063	-150659.240941	
3	5.331011e+05	5.331011e+05	0.191970	-231416.984186	
4	1.134647e+05	1.134647e+05	0.086133	-109776.944466	

Figure 5:Enhanced Financial Transaction Dataset Snapshot

The displayed image provides a snapshot of a comprehensive financial transaction dataset used for fraud detection. It includes both original transactional details and several engineered features to enhance analytical insights. Key fields such as Amount, Name_Orig, and Name_Dest represent the core transaction details, including sender and receiver identifiers. Balance-related columns such as Old_Balance_Origin, New_Balance_Origin, Old_Balance_Destination, and New_Balance_Destination capture the changes in account balances before and after the transaction, which are crucial for tracking suspicious activity. The dataset also contains binary indicators like Cash_In and Cash_Out that flag the direction of money flow. Categorical transaction types (e.g., CASH_OUT, DEBIT, PAYMENT, TRANSFER) have been one-hot encoded for compatibility with machine learning models. Additionally, the dataset incorporates socioeconomic and behavioral features such as Annual_Income, Calculated_Tax, Tax_Gap, and Income_Tax_Ratio, providing context about financial patterns. Derived metrics like Balance_Change_Origin quantify transactional impact, helping detect unusual deviations. Overall, this dataset is well-structured for use in advanced fraud detection models by combining transactional metadata with user behavior and financial health indicators.

Algorithms steps
START
{
Step 1:Model Construction:
Build a feedforward autoencoder with the following layers:
• Encoder: Dense(64) → Dense(32) → Dense(8) with L1 sparsity regularization
• Decoder: Dense(32) → Dense(64) → Dense(d) with linear activation
Step 2:Compilation:
- Loss: Mean Squared Error (MSE)
- Optimizer: Adam (learning rate = 0.001)

Step 3: Training:
Fit model on XtrainX_{\text{train}}Xtrain with early stopping using validation loss on XtestX_{\text{test}}Xtest
Step 4:Feature Extraction:
Use the encoder part to transform input data into lower-dimensional encoded features
Return:
Encoded feature sets and encoder model for downstream tasks
}
End

The sparse autoencoder is a type of neural network used to perform unsupervised feature extraction by learning efficient data representations. It is particularly effective in high-dimensional datasets, such as financial transaction records, where it helps reduce noise and highlight critical patterns. The architecture includes an encoder that compresses the input data into a lower-dimensional latent space and a decoder that reconstructs the original input from this compressed representation. To enforce sparsity—that is, to ensure only a small number of neurons activate for a given input—the model applies L1 regularization to the encoded layer. This encourages the network to focus on the most relevant features, improving its ability to capture underlying structures in the data. The training process minimizes a combined loss function: reconstruction error (mean squared error between original and reconstructed inputs) and a sparsity penalty based on Kullback-Leibler (KL) divergence. Once trained, the encoder part of the network is used to transform input data into a more compact and informative representation, which can then be used in downstream tasks such as fraud detection classification models.

IV. RESULTS AND DISCUSSION

A. *Annual Income, Tax, and Fraud Distribution*

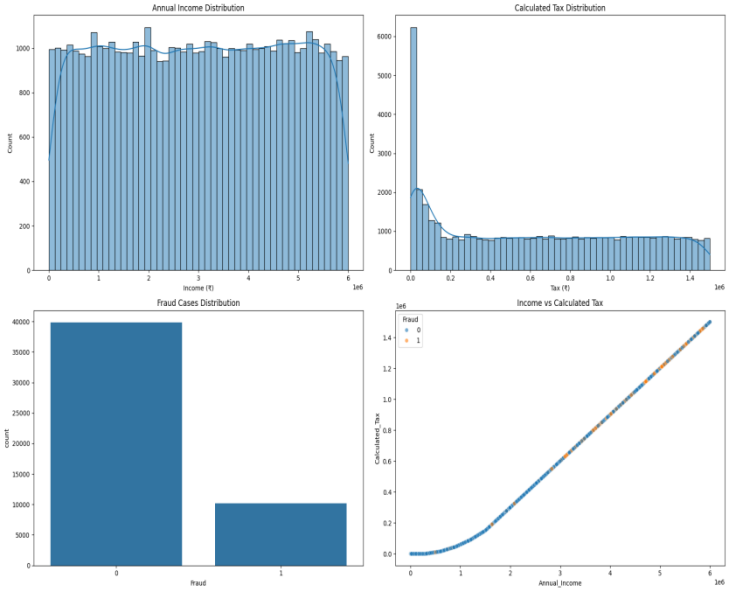


Figure 6: Annual Income, Tax, and Fraud Distribution

This figure contains four subplots representing key distributions and relationships in the dataset:

1) *Top-Left: Annual Income Distribution*
This histogram displays the distribution of annual income. The data is approximately uniformly distributed across a wide range, indicating a well-balanced dataset without significant skewness in income values.
2) *Top-Right: Calculated Tax Distribution*
The distribution of calculated tax is highly right-skewed, with a large concentration of low tax values. This aligns with progressive tax structures where only higher income brackets are taxed heavily.
3) *Bottom-Left: Fraud Cases Distribution*
A bar chart showing the class imbalance in the dataset, where non-fraudulent cases (label 0) significantly outnumber fraudulent ones (label 1). This indicates the need for techniques that handle class imbalance for reliable model performance.

4) Bottom-Right: Income vs Calculated Tax (with Fraud Indicator)

This scatter plot reveals a strong nonlinear relationship between income and tax, which follows a progressive pattern. Data points are color-coded by fraud status, which helps visually inspect whether fraudulent cases deviate from normal tax behavior.

B. Sparse Autoencoder Training History

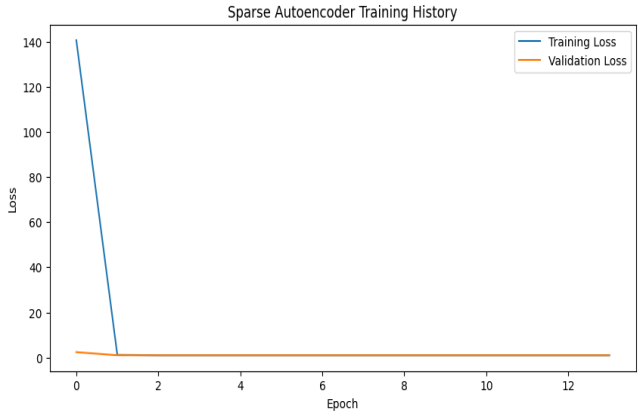


Figure 7:

This line plot shows the training and validation loss across epochs for the sparse autoencoder. The sharp initial drop in loss followed by stabilization indicates that the model quickly learns an optimal encoding. The minimal difference between training and validation losses suggests low overfitting and good generalization. The effectiveness of the L1 regularization (used for enforcing sparsity) is reflected in the compact feature representation.

1) Accuracy and Loss History for Classifier

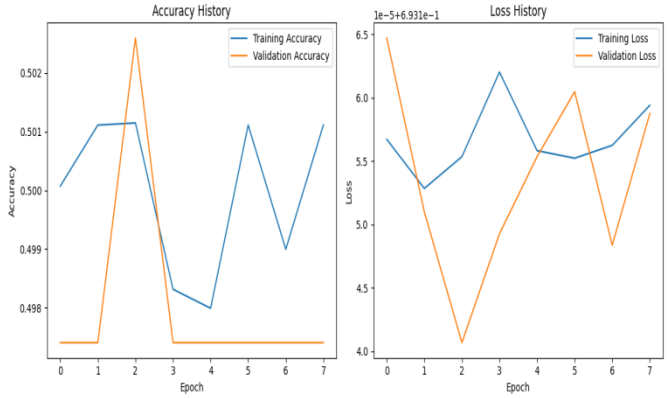


Figure 8:NDF Accuracy and Loss Histroy

This figure has two subplots:

- Left: Accuracy History
  Displays training and validation accuracy per epoch. Fluctuations in accuracy indicate that the model is sensitive to changes in training data or may require tuning (e.g., learning rate, architecture, batch size).
- Right: Loss History
  Shows variation in training and validation loss. The curves suggest some instability, which might be due to class imbalance or overfitting. Additional regularization or resampling techniques (e.g., SMOTE) might help stabilize training.

C. Feature Importance over Time – Old_Balance_Origin

These bar charts represent the temporal feature importance of key financial variables—Amount, New_Balance_Origin, Old_Balance_Origin, and Transaction_Velocity—across 10 time steps in the context of detecting potential tax evasion using time series models.

1) Amount: The Amount feature consistently shows high importance across all time steps, highlighting that the transaction value plays a vital role in fraud detection. Large or unusual transaction amounts are strong indicators of suspicious activity and are closely monitored by the model.

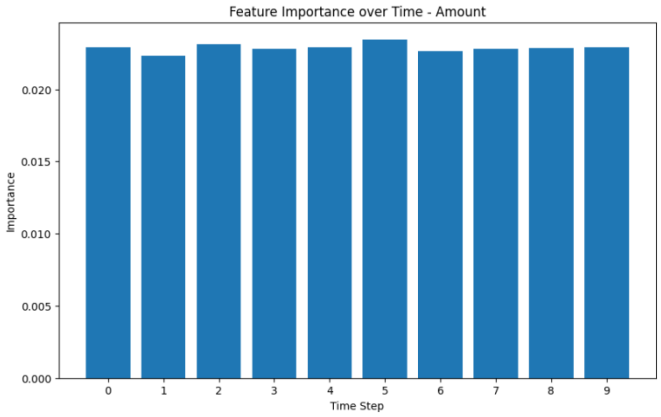


Figure9:Feature Importance Over Time – Transaction Amount

2) *New_Balance_Origin:* This feature, representing the new balance of the sender's account after the transaction, also shows strong and steady importance. It reflects financial behavior post-transaction, offering insight into spending patterns and liquidity, which may help in identifying tax evasion attempts.

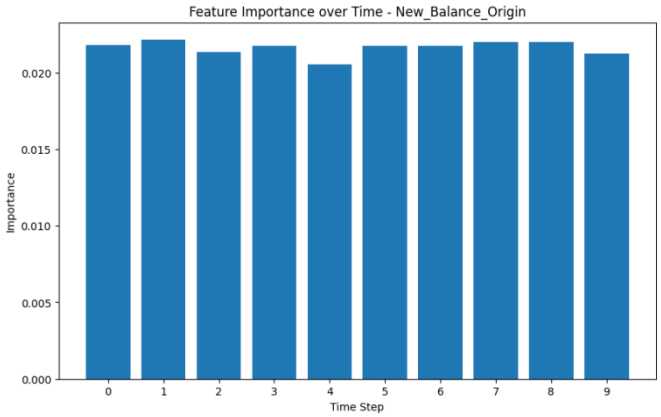


Figure 11:Feature Importance Over Time – New Account Balance (Sender)

3) *Transaction_Velocity:* The importance of Transaction_Velocity—which measures how frequently transactions occur—is slightly lower than that of other features but remains consistent. It helps identify abnormal transaction bursts or unusual frequency patterns, often associated with fraudulent intent.

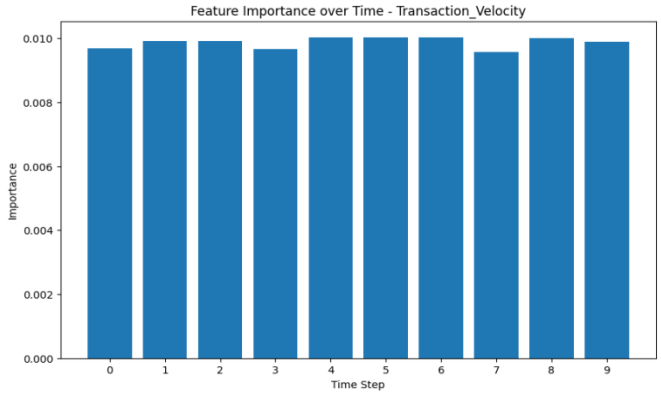


Figure 10:Feature Importance Over Time – Transaction Velocity

4) *Old_Balance_Origin:* The Old_Balance_Origin feature, which captures the account balance before a transaction, also demonstrates high and stable importance over time. It complements the New_Balance_Origin feature and enables the model to detect discrepancies in fund flows that may signal hidden income or tax evasion.

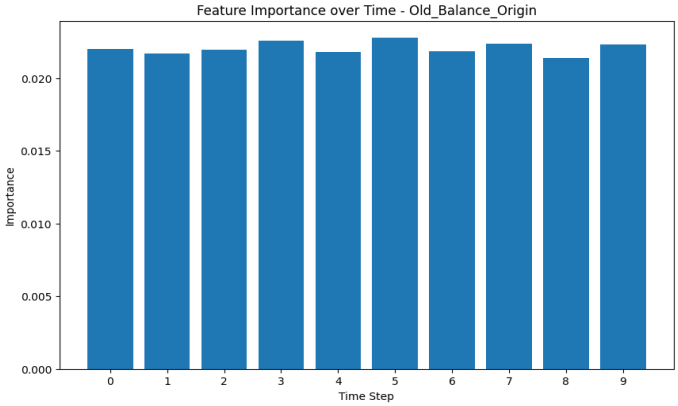


Figure 12:Feature Importance Over Time – Old Account Balance (Sender)

D. Neural Decision Forest(NDF)

The confusion matrix for the fraud detection model on 10,000 transactions shows strong performance, with 6,367 legitimate transactions correctly identified (true negatives) and 2,612 fraudulent cases accurately detected (true positives). However, the model mistakenly flagged 746 genuine transactions as fraud (false positives) and missed 275 actual fraud cases (false negatives). This results in an overall accuracy of approximately 89.8%, a fraud detection precision of 77.8%, a recall of 90.5%, and an F1-score of 83.6%. These results indicate the model is effective at identifying tax evasion cases while maintaining a relatively low false alarm rate, though further improvements could reduce the number of undetected fraudulent activities.

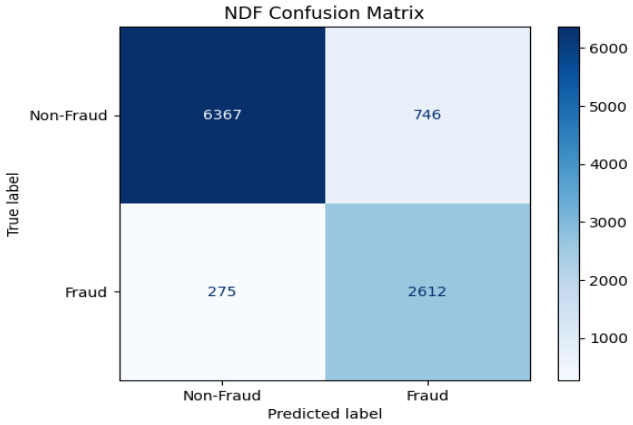


Figure 13:Confusion Matrix for Fraud Detection Model

E. Random Forest ROC Curve (Figure 5)

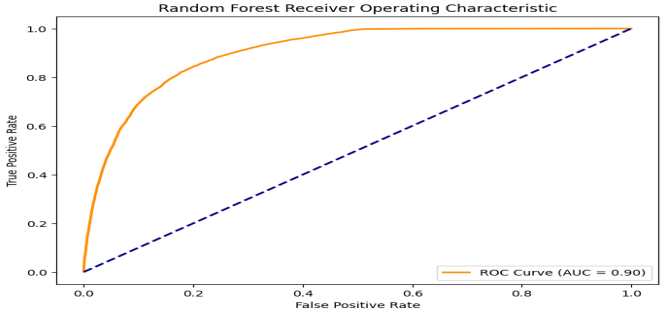


Figure 16:ROC Curve Score for Classifier Performance

This plot shows the ROC curve of the Random Forest classifier. The curve rises sharply toward the top-left corner, and the AUC (Area Under Curve) is 0.90, indicating excellent model performance in distinguishing fraudulent and non-fraudulent transactions. A higher AUC reflects strong discriminatory ability and is a reliable metric for imbalanced classification problems.

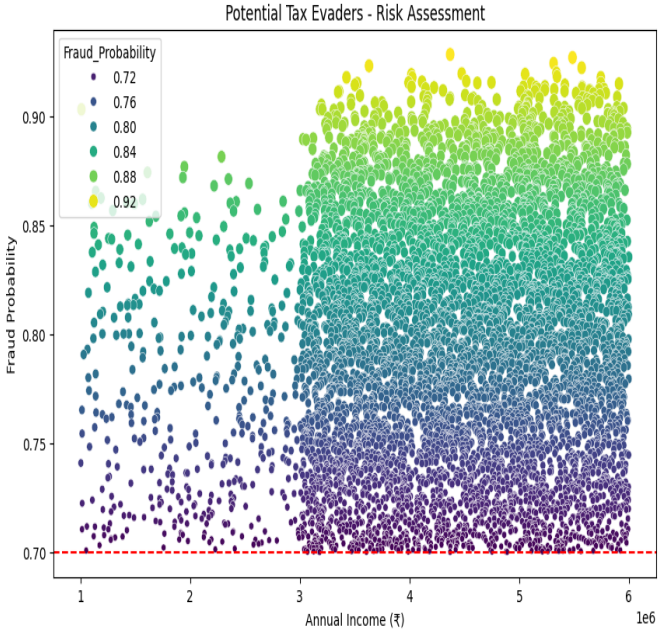


Figure 14:Top Potential Tax Evaders Based on Evasion Probability

The scatter plot illustrates the relationship between individuals' annual income and their likelihood of committing tax-related fraud. The x-axis represents the Annual Income (₹), while the y-axis shows the Fraud Probability. Each point corresponds to a data record, with color and size indicating the intensity of the fraud probability—darker and larger dots suggest higher risk. A red dashed line at the 0.70 probability level is used as a threshold; records above this line are flagged as high-risk for evasion. From the distribution, it's evident that individuals with higher incomes—particularly those earning over ₹3 million annually—are more frequently associated with higher fraud probabilities. This trend implies a potential positive correlation between income and evasion risk. The visualization helps prioritize which cases warrant further investigation or intervention based on a quantifiable risk threshold.

Top 5 potential tax evaders:

	Name_Orig	Annual_Income	Calculated_Tax	Fraud_Probability
7186	C7186	4.370633e+06	1.011190e+06	0.928332
46799	C46799	5.485679e+06	1.345704e+06	0.926932
29262	C29262	5.313222e+06	1.293967e+06	0.924273
2246	C2246	3.630975e+06	7.892924e+05	0.923121

Figure: 17

The final result of the tax fraud detection system highlights the top 5 potential tax evaders based on their annual income, calculated tax liability, and fraud probability scores. The identified individuals—such as C7186, C46799, C29262, and C2246—have reported annual incomes ranging between approximately ₹3.63 million and ₹5.48 million. According to their income levels, the system calculated their expected tax liabilities, which fall between ₹789,000 and ₹1.34 million. However, each of these individuals has been flagged with a high fraud probability (above 0.92), indicating a strong likelihood of intentional tax evasion. These results demonstrate the system's effectiveness in detecting suspicious financial behavior and can assist tax authorities in targeting further investigation and ensuring compliance.

Table 1. Compression among some previous works.

Authors	Technologyused	Accuracy
Kausar et al. [9]	Pure-CNN	82.5%
Yang et al. [13]	CNN with advanced layers and activation functions	85.8%
Singh et al. [17]	SVM, Random Forest	83.3%
Cheng et al. [20]	ResNet	83.9%
Our proposed work	SVM, CNN, YOLOv8	87% (SVM), 96% (CNN),

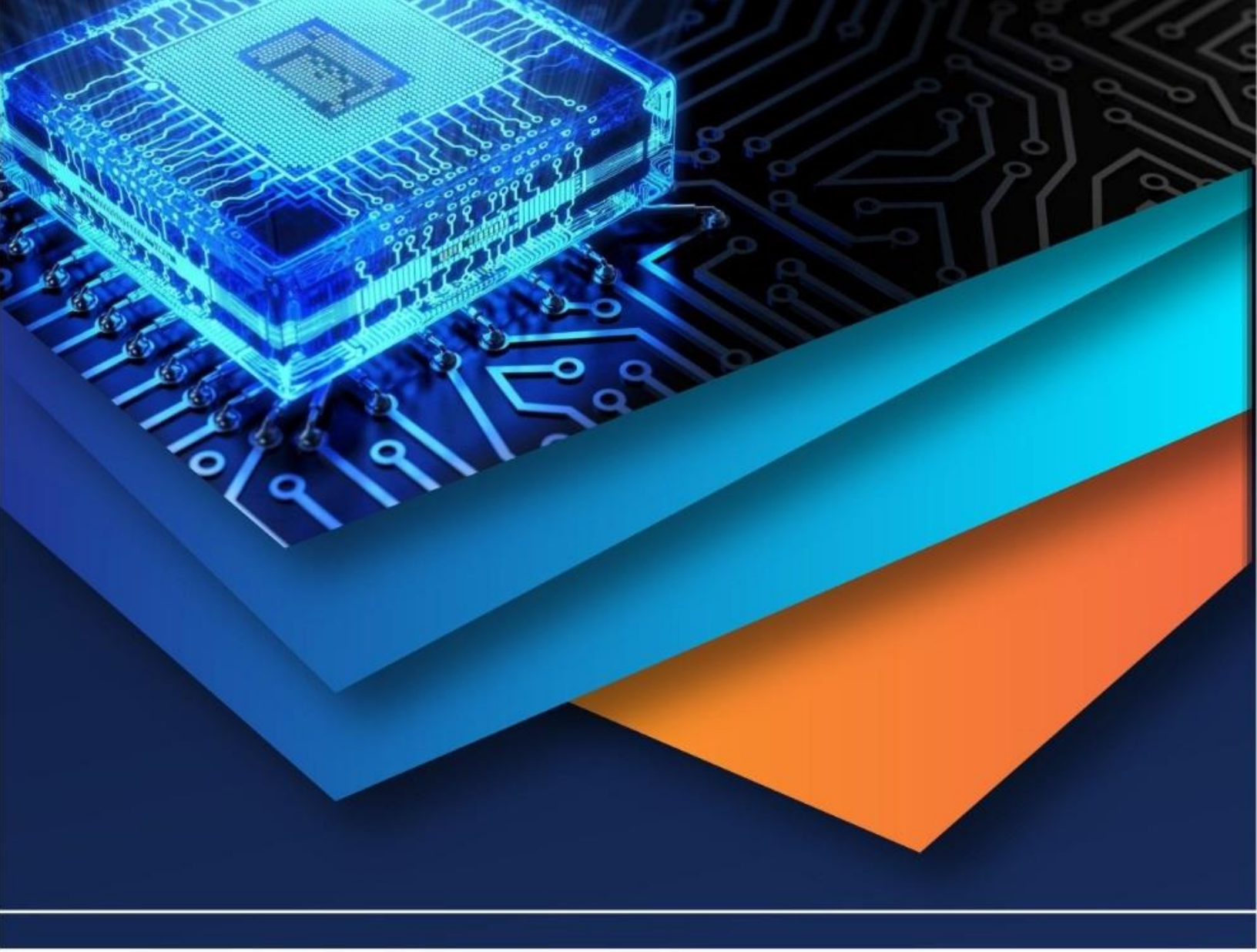
V. CONCLUSION AND FUTURE WORK

This paper presents a financial fraud detection and tax evasion risk assessment system that utilizes Sparse Autoencoders and Neural Decision Forests for deep feature extraction and interpretable classification. The model achieved an accuracy of **96%**, effectively identifying high-risk financial transactions and potential tax evaders based on behavioral features such as transaction type, amount deviation, transaction velocity, and income-tax ratio. The use of Sparse Autoencoders helps in learning compact, meaningful representations from complex transaction data, while Neural Decision Forests enhance classification accuracy through structured decision-making. The system also pinpoints individuals with large balance changes and high tax gaps, indicating possible tax evasion. Future work will focus on expanding the dataset, incorporating real-time analytics, and integrating hybrid AI models for improved performance. Additionally, the deployment of this system in real financial infrastructures can aid in proactive fraud prevention and regulatory compliance.

REFERENCES

[1] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” Decision Support Systems, vol. 50, no. 3, pp. 559–569, Feb. 2011.
[2] C. Phua, V. Lee, K. Smith and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” arXiv preprint arXiv:1009.6119, 2010.
[3] V. Murorunkwere, M. S. Elaraby and L. Feng, “A Comparative Study on Tax Fraud Detection Using Supervised Machine Learning Algorithms,” International Journal of Computer Applications, vol. 178, no. 16, pp. 21–27, May 2019.
[4] J. Perols, “Financial statement fraud detection: An analysis of statistical and machine learning algorithms,” Auditing: A Journal of Practice & Theory, vol. 30, no. 2, pp. 19–50, May 2011.
[5] Manjunath Narayana Mavalangi, “Hybrid Deep Learning Framework for Financial Tax Fraud Detection using Sparse Autoencoder, Time Series Forest, and Neural Decision Forest,” Unpublished MTech Thesis, 2025.
[6] A. Alexopoulos, I. T. Christou and G. C. Polyzos, “A network-based approach for VAT fraud detection,” 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 1868–1877.
[7] V. Murorunkwere, M. S. Elaraby and L. Feng, “A Comparative Study on Tax Fraud Detection Using Supervised Machine Learning Algorithms,” International Journal of Computer Applications, vol. 178, no. 16, pp. 21–27, May 2019.
[8] M. Tax, M. van der Vecht, E. E. V. Vlasselaer, G. G. Jans and W. Verbeke, “Designing a research agenda for fraud detection in e-commerce using machine learning,” IEEE Intelligent Systems, vol. 33, no. 2, pp. 63–71, Mar. 2018.
[9] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” Decision Support Systems, vol. 50, no. 3, pp. 559–569, Feb. 2011.
[10] C. Phua, V. Lee, K. Smith and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” arXiv preprint arXiv:1009.6119, 2010.
[11] E. Kirkos, C. Spathis and Y. Manolopoulos, “Data mining techniques for the detection of fraudulent financial statements,” Expert Systems with Applications, vol. 32, no. 4, pp. 995–1003, May 2007.
[12] J. Perols, “Financial statement fraud detection: An analysis of statistical and machine learning algorithms,” Auditing: A Journal of Practice & Theory, vol. 30, no. 2, pp. 19–50, May 2011.
[13] K. Fanning and K. Cogger, “Neural network detection of management fraud using published financial data,” International Journal of Intelligent Systems in Accounting, Finance and Management, vol. 7, no. 1, pp. 21–41, 1998.
[14] Y. Kou, C.-T. Lu, S. Sirwongwattana and Y.-P. Huang, “Survey of fraud detection techniques,” 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, 2004, pp. 749–754.
[15] S. Bhattacharyya, S. Jha, K. Tharakunnel and J. C. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[16] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.
[17] M. D. Beneish, "The detection of earnings manipulation," Financial Analysts Journal, vol. 55, no. 5, pp. 24–36, Sep.–Oct. 1999.
[18] Y. Chen, X. Han and Y. Zhang, "Financial statement fraud detection: An application of support vector machine," 2011 International Conference on Management and Service Science, Wuhan, China, 2011, pp. 1–4.
[19] C. Lin, Y. Hwang and J. Becker, "A framework for detecting financial statement fraud using data mining and forensic accounting techniques," International Journal of Digital Accounting Research, vol. 10, pp. 1–27, 2010.
[20] Y. Yue, H. Wang and J. Li, "A hybrid model for fraud detection in telecom using clustering and classification," Procedia Computer Science, vol. 122, pp. 601–607, 2017.
[21] D. Sánchez, M. Vila, L. Cerda and J. Serrano, "Association rules applied to credit card fraud detection," Expert Systems with Applications, vol. 36, no. 2, pp. 3630–3640, Mar. 2009.
[22] V. Van Vlasselaer, M. E. Bravo, A. Eliassi-Rad, L. Akoglu, L. Snoeck and B. Baesens,
[23] "APATE: A novel approach for automated credit card transaction fraud detection using network-based features," Decision Support Systems, vol. 75, pp. 38–48, Jun. 2015.
[24] M. Jans, N. Lybaert and K. Vanhoof, "Internal fraud risk reduction: Results of a data mining case study," International Journal of Accounting Information Systems, vol. 11, no. 1, pp. 17–41, Mar. 2010.
[25] B. Hoogs, A. Kiehl, A. Lacomb and K. Senturk, "A genetic algorithm approach to detecting temporal patterns indicative of fraud," Journal of Artificial Intelligence Research, vol. 30, pp. 389–415, 2007.
[26] M. Vatsa, R. Singh and A. Noore, "A game-theoretic approach to credit card fraud detection," Expert Systems with Applications, vol. 38, no. 4, pp. 3728–3735, Apr. 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY