Abstract— Gaining the empathy and trust of customers is paramount in the financial domain. However, the recurring occurrence of fraudulent activities undermines both of these factors. ATM fraud is a prevalent issue faced in today’s banking landscape. The critical challenges in fraud datasets are highly imbalanced datasets, evolving fraud patterns, and lack of explainability. In this study, we handled these techniques on an ATM transaction dataset collected from India. In binary classification, we investigated the effectiveness of various oversampling techniques, such as the Synthetic Minority Oversampling Technique (SMOTE) and its variants, Generative Adversarial Networks (GAN), to achieve oversampling. Gradient Boosting Tree (GBT), outperformed the rest of the techniques by achieving an AUC of 0.963, and Decision Tree (DT) stands second with an AUC of 0.958. In terms of complexity and interpretability, DT is the winner. Among the oversampling approaches, SMOTE and its variants performed better. We incorporated explainable artificial intelligence (XAI) and Causal Inference (CI) in the fraud detection framework and studied them via various analyses. Further, we provided managerial impact.

Keywords— Fraud detection; XAI; Causal Inference; SMOTE; GAN.

I. INTRODUCTION

Banks and Financial organizations equip their customers with diverse payment services, making their lives more comfortable while performing day-to-day transactions. Among them, the Automated Teller Machine (ATM) is one of the most widespread modes of payment services. Since then, ATMs have been facing a massive menace from fraudsters, and there is an increasing concern over these financial frauds. These fraudulent transactions inflict severe financial losses amounting to billions of dollars. Further, this will decrease the customers’ empathy for the respective banks. Due to the recent surge in the importance of fraud detection, several methods have been enforced in recent years. These mechanisms are not straightforward and involve numerous challenges in implementing a sophisticated ATM fraud detection methodology. One of the critical challenges that these financial organizations face is that they need end-to-end visibility into payment applications. Moreover, the transactions are highly skewed/imbalanced datasets. Further, explaining why and how the model has taken the decision and under what circumstances the predictions will be affected also needs to be discussed. All the acronyms used in the study are presented in Table I.

The challenges mentioned above can be mitigated by employing various data mining techniques and following the Cross-industry standard process for data mining (CRISP-DM) [9] methodology. Further, it is very important to elucidate the identification of fraudulent transactions to top management and incorporate this understanding into policy design is of the utmost important. By virtue of this, the transparency enables among stakeholders and makes them understand the rationale behind the decision-making process. We proposed an explainable and causal inference based fraud detection framework to address this challenge. The current study analyzed the ATM transaction fraud detection data collected from India.

Our salient contributions in this study are as follows:

- The data imbalance is handled using SMOTE [29-31] and GAN [32-33] and its variants.
- Most importantly, XAI [4] component is included which explain the behaviour of the binary classification techniques.

### TABLE I. ACRONYMS USED IN THE CURRENT STUDY

| Acronym | Full form |
|---------|-----------|
| ML      | Machine Learning |
| ATM     | Automated Teller Machine |
| NB      | Naive Bayes |
| LR      | Logistic Regression |
| SVM     | Support Vector Machine |
| RF      | Random Forest |
| DT      | Decision Tree |
| MLP     | Multi-Layer Perceptron |
| GBT     | Gradient Boosting Tree |
| SMOTE   | Synthetic Minority Oversampling Technique |
| SMOTE+  | Synthetic Minority Oversampling Technique – Earliest |
| ENN     | Near Neighbour |
| ADASYN  | Adaptive Synthetic Sampling approach |
| GAN     | Generative Adversarial Network |
| V-GAN   | Vanilla GAN |
| W-GAN   | Wasserstein GAN |
| XAI     | Explainable artificial intelligence |
| CI      | Causal Inference |
| CF      | Counterfactual |
| CRONYM  | CRONYM |
| CRONYM  | Cross-industry standard process for data mining |
| AUC     | The area under the receiver operator characteristic curve |
| CV      | Cross Validation |
| CR      | Classification Rate |
| TP      | True positive |
| FP      | False positive |
| TN      | True Negative |
| FN      | False Negative |
| SHAP    | SHapley Additive exPlanations |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
• We further employed Causal Inference (CI) [5] and analyzed the counterfactuals, which provide more insights into predictions as to which features and conditions critically influence the predictions.

The remainder of the paper is organized as follows: Section 2 discusses the literature review. Section 3 presents the proposed framework for ATM transactional data. Section 4 describes the dataset. Section 5 discusses the results. Finally, section 6 concludes the paper.

II. LITERATURE REVIEW

Data mining algorithms play a quintessential role in detecting fraud post-facto [6]. It can also be seen in near real-time when advanced techniques, including semi-supervised and unsupervised, are employed to mine the data in a stream [1,7]. The advanced techniques typically include online classification, online clustering, and outlier detection for anomaly detection. A reasonable amount of research is reported concerning credit card fraud detection [6,8,20]. Detecting ATM/Debit card fraud is seldom the researched topic. Only one research article dealing with ATM transactional fraud appeared, where a specialized language is proposed for proactive fraud management in financial data streams [7]. Doorronsoro et al. [10] proposed an online system for credit card fraud detection based on the neural classifier to meet real-time analytics. Chan et al. [11] pioneered a scalable solution framework for credit card fraud detection using distributed environments. Chen et al. [12] proposed a binary support vector machine combining SVM and a genetic algorithm (GA). Quah et al. [13] proposed an online detection algorithm for real-time fraud detection where the SOM is employed to detect customer behaviour and fraudulent credit card transactions. Wei et al. [14] observed various characteristics and challenges in detecting card fraud. Srivastava et al. [3] proposed a solution for credit card fraud detection using the Hidden Markov Model. Zaslavsky and Strizhak [15] applied the SOMs for credit card fraud detection. Sanchez et al. [16] applied association rule mining to identify the anomaly behaviour pattern. Syeda et al. [17] developed a parallel granular neural network in a shared memory architecture to address credit card fraud detection. Wang et al. [2] proposed an IT-based framework to address various frauds in the case study: Taiwan financial frauds. Ravisankar et al. [18] proposed a fraud detection framework to determine the critical financial ratios obtained by analyzing the financial statements. Fraquad et al. [19] proposed a modified active learning framework where the SVM is employed to extract the if-then rules and apply them to solve churn prediction and insurance fraud detection. Sundarkumar and Ravi [20] analyzed the dataset available in the CSMINING group. After extracting the features from the API calls, they invoked mutual information as the feature selection criteria were applied. Recently, Wu et al. [36] proposed an advanced management information system by employing machine learning techniques. Lai et al. [37] proposed a novel deep mixture model-based consumer fraud detection using a text convolution neural network.

Our current study is different from extant as follows:

• None of the prior works incorporated the XAI component in the fraud detection frameworks. This helps to understand the behaviour of these techniques and how the transactions are classified.

• None of the above works employed Causal Inference as a component involving the generation of counterfactuals. This provides more insights into data, such as which features and conditions critically influence the predictions.

Owing to the importance of XAI and CI, our study focuses on these aspects of studying the ATM fraud detection problem.

III. PROPOSED METHODOLOGY

This section discusses the proposed methodology in XAI and CI-based binary classification, which is depicted in Fig. 1. It includes three major phases: (a) Firstly, we discussed the context of binary classification modeling. (b) Evaluating the performance of the models. (c) Post-hoc analysis includes the CI, and the counterfactuals are studied using the chosen technique. Additionally, XAI is employed in the thus-built techniques to analyze the feature importance and their impact.

A. Fraud detection Techniques

We collected ATM transactional data from India, and the methodology adopted in this phase is depicted in Fig. 1. Indian Banks monitor all ATM transactions. The dataset consists of a few days of operations resulting in a million transactions in three modes: ATM transaction data, POS, and Internet transactions. This data comprises critical information concerning the type of transaction, mode of payment, customer details, transaction amount, etc. We masked the feature names to preserve the customers’ integrity and privacy. Data cleansing phase corrected the illogical and corrupted data, removed duplicate transactions, etc. The raw data extracted from the banks a finer data cleansing resulted in a better-quality dataset. This step is crucial because ‘garbage data in results in garbage output’.

The following steps are carried out in this phase:

• All the duplicate transactions are filtered out.

• If a feature has a percentage of null values, these features are dropped off. We did not employ any imputation technique here because we observed that a few features had almost more than 90% null values.

The data will be well-prepared for gaining insights upon completing the steps presented above. Now, we analyze the data using various data visualization techniques. This step is crucial to ascertaining trends and patterns or checking assumptions with the help of a generated statistical summary. The numerical features are subjected to Normalization. As mentioned earlier, we renamed these features to maintain data integrity and privacy. After completing the above steps, the dataset is partitioned into training and test sets in the ratio 80:20. We followed the stratified random sampling hold-out method while performing this step. Here, stratified random sampling ensures that the same proportion of both positive class and negative classes is maintained in both training and test
datasets. After performing the above steps, one must check whether the dataset is observed to have an imbalance. If the above condition is satisfied, one should apply the resampling techniques such as SMOTE [29-30], ADASYN [31], V-GAN [32], and W-GAN [33]. Now, the oversampling is performed on the training dataset by concentrating on oversampling the minority class of the observed fraudulent transactions. Here, the test dataset is kept intact because it represents reality.

B. Performance Measures

After completing the above step, we build an ML technique in this phase. This study incorporated machine learning techniques such as NB [21], LR [22], SVM [23], etc. (refer to Section 4.1). The performance of the technique is highly affected by hyperparameters. Hence, we employed the grid search hyper-parameter technique while building the technique. Furthermore, we employed the 10-Fold Cross-Validation technique and compared the performance of the machine learning techniques. Once the above process is completed, we report the measures such as AUC, Sensitivity, and Specificity. The complete details of these metrics will be discussed in Section IV.

C. Post-hoc Analysis

In the proposed methodology, this is the phase where the XAI and CI-based analysis is performed on the thus-built best-performing ML technique.

1) XAI

Often, fraud detection frameworks ignore the explainability aspect. However, XAI occupies its significance as follows: understanding how the machine learning techniques behave, how they predict the predictions, etc. This indeed increases trust and confidence. There are various ways of studying the XAI. Among them, the most popular ones are (i) local explanations and (ii) global explanations. Among them, SHapley Additive exPlanations (SHAP), proposed by Lundberg and Lee [4], is one of the most popular ones, which computes the Shapley values. It follows a game theoretic approach to explain the output of any ML technique. Hence, it is model agnostic.

The primary objective of SHAP is to explain the prediction of an instance \( x \) by determining the contribution of each feature to the prediction. This is accomplished by combining optimal credit allocation with localized explanations, utilizing the established SHAPley values from game theory. The feature values of an instance are treated as players forming a coalition, resulting in a fair distribution of the prediction payout among the feature values. These explanations are represented using an additive feature attribution method akin to a linear model. The mathematical representation of SHAPley is defined as given in Eq. (1).

\[
g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j
\]

where \( g \) is the explanation model, \( z' \in \{0,1\}^M \) is the coalition vector, \( M \) is the maximum coalition size, and \( \phi_j \in \mathbb{R} \) is the feature attribution for feature \( j \). Essentially, these SHAPley values are the average expected marginal contribution when all combinations are considered.

2) Causal Inference

It allows the researchers and practitioners to derive conclusions based on the data [5]. Thus, derived conclusions are called 'causal conclusions', which refer to the effect of the causal variable (or treatment). Studying these interventions is critical in applications such as the medical domain, financial domain, etc. Here, understanding the effect of varying the causal variable (or treatment) is very important. As depicted in Fig. 1, we incorporated the causal inference component as the post-hoc analysis into the fraud detection framework.

It generates the counterfactuals by perturbing the given original data. These samples also act as adversarial samples, which could affect the performance of the ML model. Further, while generating counterfactuals, it is important to consider the following two aspects: (i) proximity and (ii) diversity. The proximity refers to how close the generated samples are to the original data. Diversity refers to how diverse the counterfactuals are being generated, affecting the outcome. The combined loss function for the CFs is presented in Eq. (2).
\[ C(x) = \arg \min_{c_1, c_2, \ldots, c_k} \left\{ \frac{1}{k} \sum_{i=1}^{k} y \text{loss}(f(c_i), y) + \lambda_1 \sum_{i=1}^{k} \text{dist}(c_i, x) - \lambda_2 \right\} \]

where \( c_i \) is the counterfactual example, \( k \) is the total number of CFs to be generated, \( f(.) \) is the ML technique, \( y \text{loss}(.) \) is a metric that minimizes the distance between the prediction of \( c_i \), and the desired outcome \( y \), \( d \) is the total number of features, \( x \) is the original input, \( \text{dpp} \text{diversity}(.) \) is the diversity metric, \( \lambda_1 \) and \( \lambda_2 \) are the hyperparameters that balance the three parts of the function.

\[ \text{TABLE II. HYPERPARAMETERS FOR ALL THE TECHNIQUES} \]

| Model  | Hyperparameters |
|--------|-----------------|
| LR [22] | regularizer: {'l1','l2','elasticnet'}; optimizer: {'newton-cg','lbfgs','liblinear'} |
| SVM [23] | regularizer: {'l1','l2'}; loss: {'hinge','squared-hinge'} |
| DT [24] | criterion: {'gini','entropy'}; maxdepth: [1,2,..10] |
| RF [25] | criterion: {'gini','entropy'}; maxdepth: [1,2,..10] |
| GBT [26] | loss: {'deviance','exponential'}; learning_rate: [0.001,0.01,0.1]; maxdepth: [1,2,..10]; estimators: [10,20,50,100,200] |
| MLP [27] | activation: {'logistic','tanh','relu'}; solver: {'adam','sgd'} |
| V-GAN [32] Discriminator: | Hidden layer 1: 128 neurons, activation function: 'leaky ReLU'; Hidden layer 2: 64 neurons, activation function: 'leaky ReLU'; Hidden layer 3: 32 neurons, activation function: 'leaky ReLU'; Hidden layer 4: 8 neurons, activation function: 'leaky ReLU'; Epochs=10,000 |
| W-GAN [33] Discriminator: | Hidden layer 1: 256 neurons, activation function: 'leaky ReLU'; Hidden layer 2: 128 neurons, activation function: 'leaky ReLU'; Hidden layer 3: 64 neurons, activation function: 'leaky ReLU'; Hidden layer 4: 32 neurons, activation function: 'leaky ReLU'; Epochs=10,000 |

IV. DATASET DESCRIPTION

In the current study, we collected ATM transactional data from India in three modes: ATM transaction data, POS, and internet transaction. To preserve data privacy and data integrity, abiding by the policies of the bank, we renamed the features. It is observed that after refining the dataset, the total number of transactions was reduced to 7,46,724. Furthermore, many null-valued and unique features are dropped off, resulting in the total features being 10. The proportion of samples in non-fraud and fraud classes is observed to be 87.80%; 12.20%. The dataset is highly imbalanced. All the following experiments are performed in 10-Fold Cross Validation (CV).

A. Evaluation Metrics

We considered the performance metrics as follows:

1) AUC

It is proved to be a robust measure while handling imbalanced datasets and is defined as the arithmetic mean of specificity and sensitivity. AUC is presented in Eq. (4).

\[ \text{AUC} = \frac{(\text{Sensitivity} + \text{Specificity})}{2} \]

where \( \text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}} \); \( \text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}} \); TP is a true positive, FN is a false negative; TN is a true negative, and FP is a false positive.

V. RESULTS AND DISCUSSIONS

The hyperparameters employed in the current study for the ML techniques are presented in Table II. All the experiments followed the data cleansing and pre-processing techniques, as mentioned in Section III. As the dataset is unbalanced dataset, AUC is given more preference when choosing the best technique. Table III presents the AUC scores obtained by ML techniques. We also presented sensitivity and specificity scores in Tables IV and V, respectively.

Table III shows that GBT obtained an AUC of 0.963 and outperformed the rest of the techniques after balancing the dataset with SMOTE-ENN. Overall, techniques employed on the dataset balanced with SMOTE-ENN outperformed the remaining balancing techniques. The next-best performing technique is also GBT, which was obtained after balancing the dataset with SMOTE-Tomek and SMOTE. Interestingly, most techniques performed equally well when oversampled the datasets with SMOTE-Tomek and SMOTE-ENN. Further, techniques combined with SMOTE-ENN and SMOTE-Tomek performed better than those of SMOTE and other sampling techniques. Hence, this owes to the fact that these two variants removed the unwanted oversampled samples, thereby improving the performance of the techniques.

The second best technique in terms of attaining AUC is DT, which obtained an AUC of 0.957, which is marginally less when compared to the best technique GBT which yielded AUC of 0.963. However, as a decision maker, one should consider not only the numerical best model but should give importance to complexity and interpretability. DT wins over GBT, in both these aspects since the former is a much less complex and white-box model. It provides ‘IF-THEN’ rules that help to understand why the fraud might happen. As mentioned earlier, we named the features \( X_1, X_2, ..., X_k \) to maintain privacy. The first split happened at \( X_1 \). It is observed that \( X_3 \) played a significant role in formulating the rules. Considering the above tie-breakers, the decision-maker is now responsible for choosing the best model.

Table IV shows that all models obtained abysmal sensitivity before balancing the dataset. DT, RF, and GBT yielded a sensitivity of 0.0, indicating that these models are ineffective in predicting fraudulent transactions. After employing any balancing techniques, all ML models showed an improvement in the sensitivity and AUC. GBT obtained the best sensitivity of 0.962, and DT stands second with a sensitivity of 0.954. As discussed earlier, the complexity and interpretability aspect also applies here.
Considering only the numerically best AUC to choose the best model often deceives the decision maker (see Table V) where several ML techniques correctly predict all non-fraudulent transactions with 100% specificity accompanied by very low sensitivity. Thus, in unbalanced dataset, DT, RF, and GBT, obtained a specificity of 1.0 but a sensitivity of 0.0. Two more models, GT and DT, obtained a specificity score of 1.0 after balancing with W-GAN. However, after balancing with the SMOTE-ENN technique, their corresponding sensitivity score is relatively less than their counterparts.

### TABLE III. AUC Scores under the 10-Fold CV Framework

| Model  | Imbalanced | SMOTE | SMOTE | SMOTE | ADAS | V. | W. |
|--------|------------|-------|-------|-------|------|----|----|
| NB     | 0.746      | 0.794 | 0.793 | 0.795 | 0.676 | 0.769 | 0.742 |
| LR     | 0.677      | 0.822 | 0.823 | 0.825 | 0.687 | 0.773 | 0.779 |
| SVM    | 0.626      | 0.823 | 0.824 | 0.825 | 0.714 | 0.374 | 0.374 |
| DT     | 0.5        | 0.935 | 0.956 | 0.958 | 0.944 | 0.907 | 0.918 |
| RF     | 0.5        | 0.921 | 0.922 | 0.924 | 0.861 | 0.897 | 0.915 |
| GBT    | 0.5        | 0.997 | 0.963 | 0.963 | 0.942 | 0.905 | 0.910 |
| MLP    | 0.625      | 0.862 | 0.800 | 0.864 | 0.793 | 0.890 | 0.898 |

### TABLE IV. Sensitivity Scores under the 10-Fold CV Framework

| Model  | Imbalanced | SMOTE | SMOTE | SMOTE | ADAS | V. | W. |
|--------|------------|-------|-------|-------|------|----|----|
| NB     | 0.786      | 0.790 | 0.797 | 0.791 | 0.542 | 0.781 | 0.726 |
| LR     | 0.427      | 0.833 | 0.832 | 0.834 | 0.659 | 0.674 | 0.681 |
| SVM    | 0.310      | 0.838 | 0.838 | 0.840 | 0.673 | 0.675 | 0.671 |
| DT     | 0.0        | 0.950 | 0.950 | 0.983 | 0.942 | 0.828 | 0.821 |
| RF     | 0.0        | 0.909 | 0.910 | 0.911 | 0.858 | 0.814 | 0.842 |
| GBT    | 0.0        | 0.960 | 0.960 | 0.962 | 0.953 | 0.825 | 0.821 |
| MLP    | 0.312      | 0.873 | 0.878 | 0.889 | 0.764 | 0.813 | 0.855 |

### TABLE V. Specificity Scores under the 10-Fold CV Framework

| Model  | Imbalanced | SMOTE | SMOTE | SMOTE | ADAS | V. | W. |
|--------|------------|-------|-------|-------|------|----|----|
| NB     | 0.128      | 0.808 | 0.808 | 0.808 | 0.811 | 0.758 | 0.758 |
| LR     | 0.927      | 0.811 | 0.811 | 0.813 | 0.715 | 0.872 | 0.876 |
| SVM    | 0.942      | 0.808 | 0.809 | 0.810 | 0.707 | 0.872 | 0.876 |
| DT     | 1.0        | 0.961 | 0.962 | 0.963 | 0.945 | 1.0  | 1.0 |
| RF     | 1.0        | 0.964 | 0.965 | 0.964 | 0.934 | 0.996 | 1.0 |
| GBT    | 1.0        | 0.964 | 0.965 | 0.964 | 0.934 | 0.996 | 1.0 |
| MLP    | 0.939      | 0.852 | 0.814 | 0.838 | 0.822 | 0.996 | 0.982 |

### A. Statistical testing of the results

We conducted the t-test at 5% level of significance and 18 (10+10-2) degrees of freedom on top-performing models obtained over 10-Fold CV to statistically infer whether the superiority of the performance of the models is purely a random chance or is due to their superior logic.

Here, we considered the top-performing models, such as GBT, DT, and RF, using the SMOTE-ENN balancing technique. Table VI infers that p-values are less than 0.05; hence, the null hypothesis is rejected, and the alternate hypothesis is accepted. We conclude that DT is more statistically significant than RF regarding AUC, and GBT is more statistically significant than DT.

### TABLE VI. Paired t-Test Results

| Model | Parameter | t-statistic | p-value |
|-------|-----------|-------------|---------|
| DT vs GBT | 8.520 | 9.87 x 10^-5 |
| DT vs RF | 27.98 | 2.72 x 10^-5 |

### B. Explainability using SHAP

After balancing the dataset with the SMOTE-ENN technique, the following analysis is provided for the best-performing model, i.e., DT. Further, it is vital to note that all these screenshots (Fig. 2) are taken from SHAP.

#### Global Feature Importance

The feature importance score represents how "important" a feature is, which aids in predicting the target variable. The importance of the feature is depicted in Fig. 2 based on the mean SHAP values. The figure shows that X3 is the more important feature, and the latter X2 and X4 follow it. It is observed that features such as X9, X6, X12, and X10 obtained almost null feature importance.

### C. Managerial Impact through Causal Inference

DiCE [28] equips the following methods, which are used to generate counterfactuals (CFs): (i) Randomized method: Here, the CFs are generated by the random generation method; (ii) KD Tree method: The top k CFs are generated by querying the KDTree of the desired class. The intuition behind this is to utilize the closest CFs and ensure feasibility. (iii) Genetic Algorithm: where the genetic algorithm is employed to generate the counterfactuals. The intuition behind using genetic algorithms is to promote the generation of diverse CFs.

We conducted three different experiments to observe the counterfactuals on the best-performing model, i.e., DT, from the perspective of managers. Through this analysis, it would be helpful for the managers to understand what makes a non-fraudulent transaction into a fraudulent transaction, adverse effects, etc. This study considers the counterfactual generation by maintaining diversity and proximity. All the screenshots of the experiments below (Table VII-XII) were captured from DiCE. A detailed analysis is provided below:

#### Experiment I: Counterfactuals when the desired class is opposite

**Query instance:** Non-fraudulent transaction

**Treatment:** The desired class is the opposite

**Constraints:** No constraints

**Proximity weightage:** 0.2

**Diversity weightage:** 5.0

**Implication:** This helps us to identify the potential features that play a key role in turning a genuine transaction into a fraudulent one.

#### a) Method: Randomized

The results corresponding to the randomized method are depicted in Table VII. The counterfactuals shown below are that, except for X1, X3, and X9, the remaining features are unchanged despite no restriction. X1 is a categorical feature in this dataset. The counterfactual set
indicates that changing the value of feature \( X_i \) from ‘type 4’ to ‘type TD’ increases the chances of turning into a fraudulent transaction. Similarly, lower values of \( X_8 \) and \( X_9 \) could potentially increase the threat of fraudulent transactions.

**TABLE VII. RANDOMIZED METHOD: THE DESIRED CLASS IS THE OPPOSITE**

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| **TABLE VIII. KDTREE METHOD: THE DESIRED CLASS IS THE OPPOSITE****

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |

**c) Method: Genetic algorithm** The counterfactuals generated by the Genetic algorithm are depicted in Table IX. Interestingly, the set of features and their corresponding behaviour is identical to that of the KDTree method, which was explained above. Hence, the discussion over this is obviated.

**TABLE IX. GENETIC ALGORITHM METHOD: THE DESIRED CLASS IS THE OPPOSITE**

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |

**TABLE X. RANDOMIZED METHOD: THE DESIRED CLASS IS THE OPPOSITE WITH CONSTRAINTS OVER A FEATURE**

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |

**TABLE XI. KDTREE METHOD: THE DESIRED CLASS IS THE OPPOSITE WITH CONSTRAINTS OVER A FEATURE**

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |

**TABLE XII. OPPOSITE WITH CONSTRAINTS OVER A FEATURE**

| Original Sample | Generated Counterfactuals |
|-----------------|---------------------------|
| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) | \( X_6 \) | \( X_7 \) | \( X_8 \) | \( X_9 \) | \( X_{10} \) | Class |
|----------------|---------------------------|
| 4               | 10                        | 10000 | 10000 | 10000 | 0          | -15 | 15.0 | 0.3 | 18  | 0   |
| **Generated Counterfactuals** | | | | | | | | | | |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |
| TD             | -                         | -     | -     | -     | -          | -   | -   | -   | -   | 1   |

Similarly, when \( X_0 \) tends to 0, most of the counterfactuals have shown that the transaction becomes fraudulent.

c) Method: Genetic algorithm Similar to that of random and KDTree methods, the counterfactuals generated by the genetic algorithm also (refer to Table XII) show that when the \( X_1 \) category is ‘type TD’. Further, the transaction could become potentially fraudulent when \( X_2, X_3, \) and \( X_6 \) values are close to 0, and \( X_8 \) becomes 1.0 from 0.0.

In summary, the important observations as follows.

a) The randomized method is straightforward and often takes less time to generate the counterfactuals. Nevertheless, as we know, these counterfactuals are not deterministic. Hence, we recommend employing this method in systems with arbitrary counterfactuals to its simplicity.

b) Conversely, counterfactuals generated by KDTree and Genetic algorithm methods are more complex. The latter is more complicated than the former in terms of computational complexity. These two methods will come in handy to preserve the domain knowledge and check the interventions employed in a specific interval.

**VI. CONCLUSION AND LIMITATIONS**

In this study, we investigated the identification of fraudulent transactions in the ATM transactions dataset collected from India. We employed binary classification models and oversampled the minority samples by using various techniques, viz., SMOTE and its variants, GAN also to achieve oversampling. Further, we employed various ML techniques where GBT outperformed the rest of the models by achieving an AUC of
0.963, and DT stands second with an AUC of 0.958. DT is the winner if the complexity and interpretability aspects are preferred to be tie-breakers. Among all the oversampling approaches, SMOTE and its variants outperformed the rest. Further, we incorporated XAI and CI in the fraud detection framework and explored it through multifarious analyses. Additionally, we illustrated managerial impact by considering different treatments and studying them thoroughly. Hence, it helps to comprehensively understand the deployed model at assorted levels while devising business strategies.

A scalable solution tailored to the demands of today's big data world could effectively address the identified issue. Further, the current study is limited to solving problems in a static environment. Hence, it can be further posed as a real-time analytics problem and solved by using streaming analytics. Kate et al. [34] recently proposed Chaotic Fin-GAN for the financial domain problems. Hence, incorporating Chaotic Fin-GAN is also a potential future work. Further, employing financial indicators [35] is another important future direction.

ACKNOWLEDGMENT

This work is sponsored by the Department of Science and Technology (DST), India.

REFERENCES

[1] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," Artificial Intelligence Review, vol. 24, no. 1, pp. 1-14, 2005.
[2] J. Wang, Y. Liao, T. Tsai, and G. Hung, "Technology-based financial frauds in Taiwan: issue and approaches," in IEEE Conference on Systems, Man and Cyberspase, Oct, pp. 1120-1124, 2006.
[3] A. Srivastava, A. Kundu, and A. K. Majumdar, "Credit card fraud detection using hidden Markov model," IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37-48, 2008.
[4] S. Lundberg and L. Su, "A Unified Approach to Interpreting Model Predictions," ArXiv, 2017.
[5] P. Judea, "Causal inference in statistics: An overview," Statistics Surveys, vol. 3, pp. 96-146, 2009.
[6] E. W. T. Ngai et al., "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," Decision Support Systems, vol. 50, pp. 559-569, 2011.
[7] M. E. Edge and P. R. F. Sampaio, "The design of FFML: A rule-based policy modelling language for proactive fraud management in financial data streams," Expert Systems with Applications, vol. 39, pp. 9966-9985, 2012.
[8] G. J. Krishna and V. Ravi, "Evolutionary computing applied to customer relationship management: A survey," Engineering Applications of Artificial Intelligence, vol. 56, pp. 30-59, 2016.
[9] CRISP-DM: https://www-the-modeling-agency.com/crisp-dm.pdf
[10] J. R. Dorronsoro, F. Ginel, Y. Sánchez, and C. S. Cruz, "Neural fraud detection in credit card operations," IEEE Transactions on Neural Networks, vol. 8, no. 4, pp. 827-834, 1997.
[11] P. K. Chan, W. Fan, A. L. Prodomidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," IEEE Intelligent Systems, vol. 14, no. 6, pp. 67-74, 1999.
[12] R. Chen, T. Chen, and C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," International Journal of Pattern Recognition and Artificial Intelligence, vol. 20, no. 2, pp. 227-239, 2006.
[13] J. T. S. Quah and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," Expert Systems with Applications, vol. 35, no. 4, pp. 1721-1731, 2008.
[14] W. Wei, L. Li, Y. Cao, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," World Wide Web, DOI 10.1007/s11280-012-0178-0, 2012.
[15] V. Zaslavskiy and A. Strizhak, "Credit card fraud detection using self-organizing maps," Information & Security, vol. 18, pp. 48-63, 2006.
[16] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, "Association rules applied to credit card fraud detection," Expert Systems with Applications, vol. 36, no. 2, pp. 360-364, 2009.
[17] M. Syeda, Y. Zhang, and Y. Pan, "Parallel granular neural networks for fast credit card fraud detection," in 2002 IEEE International Conference on Fuzzy Systems, vol. 1, pp. 572-577, 2002.
[18] P. Ravianskar, V. Ravi, G. R. Rao, and I. Bose, "Detection of Financial Statement Fraud and Feature Selection Using Data Mining Techniques," Decision Support Systems, vol. 50, no. 2, pp. 491-500, 2011.
[19] M. A. H. Farquad, V. Ravi, and S. Bapi Raja, "Analytical CRM in Banking and Finance using SVM: A Modified Active Learning Based Rule Extraction approach," International Journal of Electronic Customer Relationship Management, vol. 6, no. 1, pp. 48-73, 2012.
[20] G. G. Sundar Kumar and V. Ravi, "A new approach for credit card fraud detection using unsupervised learning," in Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-6, 2013.
[21] T. Bayes, "An Essay towards solving a Problem in the Doctrine of Chances," Philosophical Transactions of the Royal Society of London, vol. 53, pp. 370-418, 1763.
[22] D. R. Cox, "The regression analysis of binary sequences," Journal of the Royal Statistical Society: Series B Methodological, vol. 20, no. 2, pp. 215-242, 1958.
[23] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol. 20, pp. 273-297, 1995.
[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," CRC press, 1984.
[25] T. K. Ho, "Random decision forests," in Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278-282, 1995.
[26] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.
[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press, pp. 318-362, 1986.
[28] R. K. Mothilal, A. Sharma, and T. Chenhao, "Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations," ArXiv, 2019.
[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.
[30] G. E. A. P. Batista, A. L. C. Bazzan, and M. A. Monard, "Balancing Training Data for Automated Annotation of," in Proceedings of the Second Brazilian Workshop on Bioinformatics, pp. 35-43, 2003.
[31] H. He, Y. Bai, F. A. no. 2, pp. 360-364, 2009.
[32] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," CRC press, 1984.
[33] T. K. Ho, "Random decision forests," in Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278-282, 1995.
[34] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.
[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press, pp. 318-362, 1986.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.