Article

Not peer-reviewed version

Multi-Entity Real-Time Fraud Detection System using Machine Learning: Improving Fraud Detection Efficiency using FROST-Enhanced Oversampling

M PRASAD [*] and T SRIKANTH

Posted Date: 23 April 2024

doi: 10.20944/preprints202404.1461.v1

Keywords: Classification, sampling, majority class, minority class, Classifier, Smote, Frost, k-nearest neighbors, Random Forest

Article

Multi-Entity Real-Time Fraud Detection System Using Machine Learning: Improving Fraud Detection Efficiency using FROST-Enhanced Oversampling

MARADANA DURGA VENKATA PRASAD [1] and SRIKANTH T [2]

[1] Research Scholar, Department of Computer Science and Engineering, Gandhi Institute of Technology and Management (GITAM), Visakhapatnam, Andhra Pradesh, India.

[2] Associate Professor, Department of Computer Science and Engineering, Gandhi Institute of Technology and Management (GITAM), Visakhapatnam, Andhra Pradesh, India. sthota@gitam.edu

* Correspondence: powersamudra@gmail.com

Abstract: Fraudulent transactions pose a significant threat to financial institutions and e-commerce platforms.Machine learning models, trained on historical labeled data (fraudulent vs. legitimate transactions), are often employed to identify and prevent fraud. However, real-world datasets frequently exhibit class imbalance, where fraudulent transactions (minority class) are significantly outnumbered by legitimate transactions (majority class). Machine learning models may perform poorly as a result of this imbalance, underestimating fraud and favouring the majority class. This paper proposes a novel approach to address class imbalance and improve fraud detection accuracy. We explore the implementation of FROST (Feature space RObust Synthetic saTuration) oversampling, a technique specifically designed to generate synthetic samples for the minority class. The FROST function leverages the k-nearest neighbors (KNN) algorithm and a user-defined amplification factor (m) to create synthetic data points that closely resemble existing minority class instances. We integrate the FROST-enhanced oversampling technique into the machine learning pipeline for fraud detection. The paper evaluates the effectiveness of this approach compared to traditional oversampling methods and analyzes its impact on classification accuracy metrics.

Keywords: Classification; sampling; majority class; minority class; Classifier; Smote; Frost; k-nearest neighbors; Random Forest

Fraudulent activities continue to plague the financial and e-commerce sectors. Machine learning models trained on historical transaction data are a popular tool for fraud detection [1]. Class imbalance, a prevalent problem when the number of fraudulent transactions (minority class) is much smaller than the number of valid transactions (majority class), might, nevertheless, undermine the effectiveness of these models. This imbalance can lead to models that prioritize the majority class and fail to accurately detect fraud [2].

Imagine training a classifier to identify rare diseases in medical scans. If 99% of your scans are from healthy patients and only 1% show signs of the rare disease, your model is likely to struggle. This is because of a common challenge in machine learning: class imbalance.

What is Class Imbalance?

Class imbalance occurs when a dataset has a significant skew in the distribution of class labels. In our medical scan example, the "healthy" majority class vastly outnumbers the "rare disease" minority class). This imbalance can lead to several problems:

- Biased Models: Algorithms During training, machine learning algorithms frequently give priority to the majority class. In our example, the model might learn to perfectly identify healthy scans but completely miss the rare disease, leading to misdiagnoses [3].
- Poor Performance Metrics: Traditional accuracy metrics become unreliable when dealing with imbalanced classes. A high overall accuracy might mask the model's inability to detect the minority class effectively [4].

Real-World Examples of Class Imbalance:

- **Fraud Detection:** Credit card transactions are mostly legitimate (majority class), with a small number being fraudulent (minority class). A model trained on imbalanced data might miss fraudulent transactions altogether.
- **Spam Filtering:** Most emails are legitimate (majority), with a smaller portion being spam (minority). An imbalanced model might classify some legitimate emails as spam (false positives) while missing actual spam emails.
- **Customer Churn Prediction:** Most customers remain loyal (majority), with a few churning (minority). An imbalanced model might fail to identify customers at risk of churning, hindering efforts to retain them.

Addressing Class Imbalance:

A variety of strategies can be used to rectify the imbalance in classes and enhance the performance of the model.:

- **Oversampling:** Replicating data pieces from minority classes in order to improve their representation. This can be done randomly or strategically (e.g., SMOTE algorithm) [5].
- **Undersampling:** Reducing the number of majority class data points to achieve a more balanced distribution. However, this can discard valuable information [6].
- **Cost-Sensitive Learning:** During training, instances of the minority class that were incorrectly classified are given larger weights, which forces the model to focus more on the minority class. [7].
- **Hybrid Approaches:** Combining techniques like oversampling with feature selection or cost-sensitive learning can be effective.

By addressing class imbalance, you can ensure your machine learning models perform well on all classes, not just the majority. This leads to more reliable predictions and better decision-making in real-world applications.

Table 1. Class imbalance can affect a machine learning model for spam email classification. Table 1. Different types of messages in Email Communication.

Email Text	Label
Discussing upcoming meeting	Ham
Promotional offer - 50% off!	Spam
Forgot your password? Reset here.	Phishing
Important update from your bank.	Ham
Free gift card! Click here to claim.	Spam
Lunch order for tomorrow?	Ham
Win a trip to Hawaii!	Spam
Meeting reminder: 10:00 AM	Ham
Your account has been suspended.	Phishing
Update your billing information.	Phishing

Explanation:

This dataset has 10 email messages labeled as either "Ham" (legitimate) or Spam/Phishing (malicious). As you can see, there are only 3 emails classified as Spam/Phishing (minority class), while 7 are classified as Ham (majority class). This is a classic example of class imbalance.

Impact on a Machine Learning Model:

Imagine training a model on this imbalanced data. The model might prioritize learning to identify the frequent "Ham" emails and achieve a high overall accuracy. However, it might struggle to accurately classify the less frequent Spam/Phishing emails, potentially leading to:

- **Missed Spam:** The model might classify some spam emails as legitimate (false negatives).
- **Unnecessary Filtering:** The model might flag some legitimate emails as spam (false positives).

Addressing Imbalance:

Techniques like oversampling or under-sampling can be applied to balance the dataset. Oversampling could duplicate the Spam/Phishing emails to create a more even distribution. Under-sampling could reduce the number of Ham emails.

This example highlights the importance of recognizing and addressing class imbalance in machine learning. By ensuring a balanced representation of all classes, you can train models that perform well across the board and make more reliable predictions.

2. Literature Survey

Various research works that have used the oversampling techniques are discussed below which have been taken from the cutting edge implementation of the techniques.

In addressing the challenges of the imbalanced data in image analysis, a novel oversampling technique, DeepSMOTE, emerged as a tailored solution for deep learning models. Distinguished from GAN-based methods, DeepSMOTE eschews a discriminator, instead leveraging an encoder/decoder structure with a SMOTE-based loss function to generate artificial images that maintain the essence of original data. This innovation not only enriches minority classes but also offers a publicly accessible implementation, contributing to the field's advancement [8]. In the quest to enhance classification in unbalanced datasets, particularly for detecting loose particles in sealed electronics, LR-SMOTE emerges as a promising advancement. Building upon the standard SMOTE algorithm, LR-SMOTE generates new samples by gravitating towards the data's central distribution, thereby preserving its integrity and avoiding outlier interference. Empirical evidence suggests that LR-SMOTE, in tandem with algorithms like Random Forest and SVM, surpasses its predecessor in accuracy, effectiveness, and AUC metrics, marking a significant stride in the field of imbalanced data classification [9]. Addressing the pervasive challenge of imbalanced learning in data mining, SMOTE based Class-Specific Extreme Learning Machine (SMOTE-CSELM) emerged as an innovative solution. This technique, which takes inspiration from Weighted Kernel-based SMOTE (WKSMOTE), uses class-specific regularisation in conjunction with minority oversampling. The goal of SMOTE-CSELM's architecture is to reduce the bias towards majority classes by increasing the impact of minority class samples on the decision regions of classifiers. Its efficacy is validated through extensive testing on real-world datasets, showcasing its potential as a computationally efficient tool for balanced classification [10].In the quest for accurate recreational water quality prediction, the study in [11] acknowledged the challenge posed by data imbalance, particularly in Faecal Indicator Bacteria (FIB) levels. The prevalent surplus of safe readings over unsafe ones compromises the models' ability to detect hazardous water conditions. To counteract this, the study advocates the use of ADASYN, an adaptive synthetic sampling approach, to enrich the minority unsafe class data. Machine learning models that have been trained later, including KNN, boosting decision trees, and artificial neural networks, with this augmented dataset yielded promising results. Notably, all models, barring support vector machines, attained commendable accuracy and sensitivity rates, signifying their potential in reliable water quality prediction. The study highlights the superior performance of boosting decision trees and artificial neural networks, underscoring their value in safeguarding public health through enhanced water quality monitoring. Confronting the myriad of network threats, the study in [12] introduces a novel intrusion identification framework that integrates the adaptive synthetic sampling (ADASYN) algorithm with a refined convolutional neural network (CNN) model. This hybrid approach, termed AS-CNN, aims to address the deficiencies of traditional intrusion detection systems (IDSs), such as high false alarm rates and poor generalization. The ADASYN algorithm is employed to equalize sample distribution, enhancing the model's ability to recognize smaller, yet critical, attack samples. Furthermore, the study presents an improved CNN architecture that incorporates a split convolution module (SPC-CNN), designed to enhance feature diversity & reduce interchannel redundancy. The AS-CNN model's efficacy is validated on the NSL-KDD dataset, where it outperforms conventional CNN and RNN models in accuracy, detection rates,

doi:10.20944/preprints202404.1461.v1

and false alarm rates. The results indicate a substantial enhancement in network security, positioning AS-CNN as a significant advancement in the field of intrusion identification [12]. In the domain of Software Fault Prediction (SFP), the proposed research in [13] introduced a novel approach utilizing Butterfly optimization for feature selection and Ensemble Random Forest with Adaptive Synthetic Sampling (E-RF-ADASYN) for fault prediction. This method addresses challenges posed by imbalanced datasets in early-stage fault prediction and demonstrates superiority by achieving an AUC of 0.854767, outperforming the Rough-KNN Noise-Filtered Easy Ensemble (RKEE) method's AUC of 0.771 [13].

Imbalanced binary datasets (where one class has less than 40% of the data) cause bias in classification algorithms. SMOTE, a technique that generates synthetic data to balance datasets, suffers from inefficiency due to random generation. This paper [14] proposed HCAB-SMOTE, a novel approach that combines undersampling of majority noise and targeted oversampling of borderline areas using k-means clustering. HCAB-SMOTE aims to minimize generated data while maximizing classification accuracy. Experiments show HCAB-SMOTE outperforms existing methods by achieving the highest accuracy with the fewest synthetic instances. In the realm of financial institutions, imbalanced classification for bankruptcy prediction holds significant importance. Although many statistical and AI techniques have been put forth, deep learning algorithms for classification and prediction problems have seen a recent upsurge in interest. In this context, [15] introduced a novel approach, BSM-SAES, which combines Borderline Synthetic Minority oversampling technique (BSM) with Stacked AutoEncoder (SAE) using the Softmax classifier. It aimed to develop an accurate bankruptcy prediction model inclusive of feature extraction. To assess the model's performance, we compare it with traditional machine learning methods like k-nearest neighbor, decision tree, support vector machine, and artificial neural network, C5.0, on Polish imbalanced datasets. Results demonstrate the superior efficiency of proposed BSM-SAES model in predicting and classifying the financial status of firms compared to other methods [15]. This paper presents a new framework for network anomaly detection that addresses both data imbalance and feature selection. Unlike traditional binary intrusion classification, this approach tackles the challenge of multi-class network intrusion detection. A resampling approach is proposed to solve the widespread problem of imbalanced data in network intrusion datasets. This strategy combines random sampling with Borderline SMOTE, a technique for creating synthetic data points. Additionally, feature selection based on information gain rate is employed to optimize the feature set used by the model. Experiments using three machine learning algorithms (KNN, DT, RF) are conducted to identify the optimal feature selection scheme for the proposed framework [16].

This paper [17] proposed a method for emotion recognition using EEG signals, employing a CNN with Borderline-SMOTE for data augmentation. Using the DEAP dataset, EEG signals are pre-processed, and features are extracted in the frequency domain. Data augmentation ensures a balanced dataset. Results show superior performance, with average accuracy rates of 97.47% and 97.76% for valence and arousal dimensions, respectively. The inclusion of Borderline-SMOTE enhances affective emotion recognition compared to methods without it [17]. In the realm of obstetrics, assessing amniotic fluid volume is crucial for monitoring fetal development. This study [18] introduced a novel approach employing a model consisting of a convolutional neural network (CNN) for feature extraction, chi-square for feature selection, safe level synthetic minority oversampling technique (SMOTE) for data oversampling, and XGBoost for classification. Through comprehensive testing and analysis, the proposed model demonstrates superior accuracy performance, achieving 96.5% accuracy in identifying amniotic fluid volume. This outperforms previous studies and signifies advancements in the field. In the context of handling imbalanced data using SMOTE-based algorithms, selecting an appropriate value for the parameter k (number of nearest neighbors) significantly impacts classification performance. This paper [19] introduced a novel approach to suggest an optimal k value using the Natural Neighbor algorithm. Four SMOTE-based algorithms are employed to balance datasets, namely standard SMOTE, Safe-Level-SMOTE, ModifiedSMOTE, and Weighted-SMOTE. Evaluation is conducted using F-measure and Recall metrics with Support Vector Machine classifiers across six datasets with varying in ratios of imbalance. The strategy is effective in enhancing classification accuracy, as evidenced by the results, which show that the proposed k values produce classification performance that is closer to the optimum than the default k values.

This study [20] examined the efficacy of Support Vector Machine, Naive Bayes, and k-nearest neighbors classifiers in conjunction with resampling techniques (Tomek link, SMOTE and their combination) for fault classification in electrical machines. Using both simulated and experimental imbalanced data from a wound-rotor induction generator, performance metrics like precision, recall, and F1-score are employed. Results show that the combination of SMOTE with Tomek link yields the best performance across all classifiers, with the k-nearest neighbors classifier coupled with this resampling technique achieving the most accurate classification results. These findings offer valuable insights for researchers and practitioners in condition monitoring for electrical machines, especially in scenarios with limited fault data availability.

In [21], we introduce DEXGB_Glu, a method aimed at identifying lysine glutarylation sites by utilizing XGBoost as a classifier, optimized through the differential evolution algorithm. Given the imbalance between positive and negative samples, we employ the Borderline-SMOTE method to synthesize additional positive samples, aligning their quantity with negative samples. Subsequently, the Tomek links technique is utilized to filter out noise data. Our analysis reveals that the differential evolution algorithm significantly enhances performance, while the combination of Borderline-SMOTE and Tomek links effectively addresses the imbalance issue. Overall, our method outperforms existing approaches in predicting glutarylation sites. The data and code are publicly available on GitHub for further exploration and implementation. [22] applies the Smote-Tomeklink method and Random Forest algorithm to address the imbalance in the Pima Indian Diabetes dataset. By balancing the dataset and using Random Forest for classification, the approach achieves high accuracy, sensitivity, precision, and F1-score. Specifically, utilizing Smote-Tomeklink enhances Random Forest performance, achieving 86.4% accuracy, 88.2% sensitivity, 82.3% precision, and 85.1% F1-score. These results underscore the effectiveness of Smote-Tomeklink in improving classification performance in health data analysis.

Based on the previous listed works we understand that In machine learning, class imbalance is a common problem when some classes are underrepresented in the data, resulting in biassed models.

To address this, various methods of oversampling have been developed:

1. **SMOTE (Synthetic Minority Over-sampling Technique):** It is used to generate synthetic samples by interpolating between existing minority class instances [23].
2. **ADASYN (Adaptive Synthetic Sampling):** It is like SMOTE, but it produces more artificial data for harder-to-learn minority classes. [24]**.**
3. **Borderline SMOTE:** It concentrates on the minority class instances that are nearer to the borderline with the majority class [25].
4. **Safe-Level SMOTE:** Modifies the SMOTE algorithm by incorporating a safety level to prevent overgeneralization [26].
5. **SMOTE Tomek Links:** It joins SMOTE with Tomek Links, which are pairs of nearest neighbors from different classes. The Tomek Links are removed to increase the separation between classes[27].

These techniques have been used to enhance the functionality of a number of machine learning algorithms, such as Random Forest (RF). During training, RF creates a large number of decision trees and outputs the class that is the mean of the classes of each individual tree. RF is an ensemble learning technique.

The **FROST (Feature space RObust Synthetic saTuration)** technique is a newer approach that addresses class imbalance. While specific details on FROST’s application in RF are available, the technique is generally **designed** to optimize sub-sampling in a way that minimizes recovery error. For a given training set, it is a non-parametric learning algorithm that computes a small collection of optimal sample directions. Strong theoretical assurances from the compressed sensing field and notable increases in reconstruction quality over state-of-the-art techniques are two of FROST's benefits. Its speed, consistency, and ease of use in terms of implementation and theory make it a potentially better method for resolving class imbalances in machine learning algorithms such as RF.

Table 2. Different class Imbalance addressing techniques.

Technique	Description	Advantages	Technique
SMOTE (Synthetic Minority Over-sampling Technique)	It creates synthetic samples by interpolating between existing minority class instances.	Simple to implement	Can lead to overfitting
ADASYN (Adaptive Synthetic Sampling)	Focuses on generating more synthetic data for harder-to-learn minority class instances.	Addresses limitations of SMOTE	More complex to implement
Borderline SMOTE	Targets minority class instances close to the decision boundary with the majority class.	Aims to improve classification on the borders	May overfit on specific borderline regions
Safe-Level SMOTE	Introduces a "safety level" to avoid generating synthetic points too far from existing minority class instances.	Reduces overgeneralization	Requires careful selection of the safety level parameter
SMOTE Tomek Links	Combines SMOTE with Tomek Links (identifies noisy data points) to improve class separation.	Addresses noisy data along with oversampling	More complex to implement compared to basic SMOTE
FROST (Feature space RObust Synthetic saTuration)	Generates synthetic data points by amplifying the difference between a chosen feature value of a minority class instance and its neighbors.	Potentially more control over synthetic data generation	Relatively new technique, requires further research on optimal parameter settings

3. Proposed Algorithm

3.1. FROST-Enhanced Oversampling

This paper introduces FROST (Feature space RObust Synthetic saTuration) oversampling as a novel approach to address class imbalance in fraud detection. The FROST function utilizes the following steps:

Let's walk through the FROST oversampling process with a sample dataset to understand how it works:

Scenario: Imagine you're building a fraud detection model using transaction data. You have features like transaction amount, location, and time. Your minority class is fraudulent transactions, and you want to use FROST to oversample them.

1. **Choose Initial Feature (B):**

   You decide to focus on the "transaction amount" feature for oversampling (initial_feature_index).

2. **Calculate Similarity Matrix (C):**

   Suppose you have two fraudulent transactions with amounts:
   Transaction 1: $1000
   Transaction 2: $500
   Calculate the absolute difference between their transaction amounts: |$1000 - $500| = $500
   This difference represents a basic measure of similarity. You can use more complex distance metrics in the actual implementation.

3. **Identify k-Nearest Neighbors (KNN) (D):**

   Let's say k (number of neighbors) is set to 1. Since there are only two fraudulent transactions, Transaction 1 will be the nearest neighbor for Transaction 2 (and vice versa) based on their similar transaction amounts.

4. **Generate Synthetic Data Points? (E):**

   Yes, we haven't processed all minority class points yet.

5. **Calculate Difference & Amplify (F):**

   For Transaction 1, the difference between its own amount ($1000) and its nearest neighbor's amount ($500) is $500.
   Now, you define the amplification factor (m). Let's say m is set to 2 (user-defined parameter). The amplified difference becomes $500 * 2 = $1000.

6. **Create New Data Point with Amplified Difference (G):**

   Create a new synthetic data point with all the features of Transaction 1 except for the transaction amount.
   New Synthetic Transaction:
   Transaction Amount: $1000 (original amount + amplified difference)
   Location: (same as Transaction 1)
   Time: (same as Transaction 1)

7. **Add New Point to Synthetic Data Set (H):**

   Add this newly created synthetic fraudulent transaction to your dataset of minority class instances.

8. **Repeat (E-H):**

   Repeat steps E-H for Transaction 2 as well. You might calculate a slightly different amplified difference based on its nearest neighbor.

9. **End (J):**

   Once you've processed all fraudulent transactions, the FROST oversampling is complete. You now have an increased number of synthetic fraudulent transactions to improve your model's ability to learn and detect fraudulent patterns.

   Key Points:

   FROST focuses on amplifying the difference in a chosen feature to create synthetic data points that resemble existing minority class instances.

The amplification factor (m) allows you to control the extent of this change.

This method can be very helpful when handling characteristics (such transaction amount) that significantly affect fraudulence.

3.2. Methodology

Entity Relationship (ER) Modeling

The diagram elements are used to the entities and their relationships based on the CSV files you mentioned:

Entities:

1. Account (from account_activity.csv)
2. Customer (from customer_data.csv)
3. Fraud Indicators (from fraud_indicators.csv)
4. Suspicious Activity (from suspicious_activity.csv)
5. Merchant (from merchant_data.csv)
6. Transaction Category (from transaction_category_labels.csv)
7. Amount Data (from amount_data.csv)
8. Anomaly Scores (from anomaly_scores.csv)
9. Transaction Metadata (from transaction_metadata.csv)
10. Transaction Records (from transaction_records.csv)

Relationships:

- **Customer** (one-to-one) **Account**: Each customer has one account associated with them.
- **Account** (many-to-many) **Transaction Record**: An account can have many transactions, and a transaction record can be associated with multiple accounts (joint accounts).
- **Transaction Record** (one-to-one) **Amount Data**: Each transaction record has one set of amount data associated with it.
- **Transaction Record** (one-to-one) **Transaction Metadata**: Each transaction record has one set of metadata associated with it.
- **Transaction Record** (one-to-many) **Anomaly Scores**: A transaction record can have multiple anomaly scores generated by different models.
- **Transaction Record** (many-to-one) **Transaction Category**: A transaction can belong to one specific category (e.g., groceries, travel).
- **Transaction Record** (many-to-many) **Merchant**: A transaction can involve one merchant, and a merchant can have many transactions. (Consider scenarios like online marketplaces)
- **Transaction Record** (many-to-many) **Suspicious Activity**: A transaction record can be flagged for multiple suspicious activities, and a suspicious activity can be identified in multiple transactions.
- **Suspicious Activity** (many-to-many) **Fraud Indicators**: A suspicious activity can be triggered by multiple fraud indicators, and a fraud indicator can contribute to identifying multiple suspicious activities.

Cardinalities:

- **One-to-One (1:1)** - One instance of one entity and one instance of another are related to each other. (e.g., Customer - Account)
- **Many-to-One (N:1)** - A single instance of one entity is linked to several instances of another. (e.g., Transaction Record - Transaction Category)
- **Many-to-Many (N:M)** - Numerous occurrences of one entity are connected to numerous instances of another entity.. (e.g., Transaction Record - Merchant)

Class Diagram

Classes:

- **Account**: Represents a customer's financial account.
- **Customer**: Represents a customer with personal information.
- **FraudDetectionSystem**: Orchestrates the fraud detection process.
- **SuspiciousActivityManager**: Manages the identification and flagging of suspicious transactions.
- **TransactionProcessor**: Processes incoming transaction data.
- **TransactionRecord**: Represents a single transaction record with details.

- **FraudIndicators**: Encapsulates rules or checks for identifying potential fraud.
- **TransactionAnalyzer**: Analyzes transaction data using various techniques.
- **AnomalyScoreCalculator**: Calculates anomaly scores based on transaction attributes.

Relationships:

- **FraudDetectionSystem**<<uses>>TransactionProcessor: The system uses the processor to handle incoming transactions.
- **FraudDetectionSystem**<<composes>>SuspiciousActivityManager: The system manages the manager component responsible for identifying suspicious activities.
- **TransactionProcessor**<<creates>>TransactionRecord: The processor creates transaction records from raw data.
- **TransactionRecord**<<associates with>> Account: A transaction record is associated with a specific account.
- **TransactionRecord**<<associates with>> Merchant: A transaction record involves a merchant.
- **TransactionRecord<<uses>>TransactionAnalyzer: The record utilizes the analyzer for in-depth analysis.**
- **TransactionAnalyzer<<uses>>FraudIndicators: The analyzer uses fraud indicators to identify potential red flags.**
- **TransactionAnalyzer<<uses>>AnomalyScoreCalculator: The analyzer uses the calculator to generate anomaly scores.**
- **SuspiciousActivityManager<<associates with>>TransactionRecord: The manager identifies suspicious activities within transaction records.**
- **SuspiciousActivityManager<<associates with>>FraudIndicators: The manager considers fraud indicators when flagging suspicious activities.**

We implement the following steps to evaluate the proposed approach:

1. **Data Acquisition:** Obtain a labeled dataset containing historical transaction data with fraudulent and legitimate transactions clearly identified.
2. **Data Preprocessing:** Clean and scale the data to ensure compatibility with machine learning models.
3. **Class Imbalance Analysis:** Calculate the dataset's degree of class imbalance.
4. **Model Training:** Train machine learning models for fraud detection with the following approaches:
   - **Baseline Model:** Trained on the original imbalanced dataset.
   - **Oversampling with Random Replication:** Traditional oversampling by replicating minority class data points.
   - **Oversampling with SMOTE:** Oversampling using the SMOTE algorithm.
   - **Oversampling with FROST:** Oversampling using the proposed FROST function with different values for k and m.
5. **Model Evaluation:** Measures of classification accuracy such as precision, recall, F1-score, and AUC-ROC should be used to assess each model's performance.

4. Results

This study investigates fraud detection in online transactions using a dataset containing transaction details, customer information, and merchant data. The dataset undergoes thorough analysis and feature engineering utilizing SMOTE (Synthetic Minority Over-Sampling Technique) and FROST (Feature-Space Oversampling Technique) techniques to address the imbalanced nature of the data. Cross-validation experiments reveal that FROST outperforms SMOTE for the given dataset, particularly in improving the minority class representation.

A Random Forest classifier is employed for the classification task, taking advantage of its capacity to handle intricate datasets and detect non-linear correlations. Hyper-parameter tuning is applied to optimize the Random Forest model's performance. The outcomes show that the Random Forest classifier obtains 100% accuracy on the dataset, which is corroborated by other evaluation metrics that surpass 95%, including precision, recall, and F1-score.

Overall, this study showcases the effectiveness of FROST in increasing the fraud detection performance in online transactions, and highlights the robustness of the Random Forest classifier when coupled with appropriate oversampling techniques and hyper-parameter tuning.

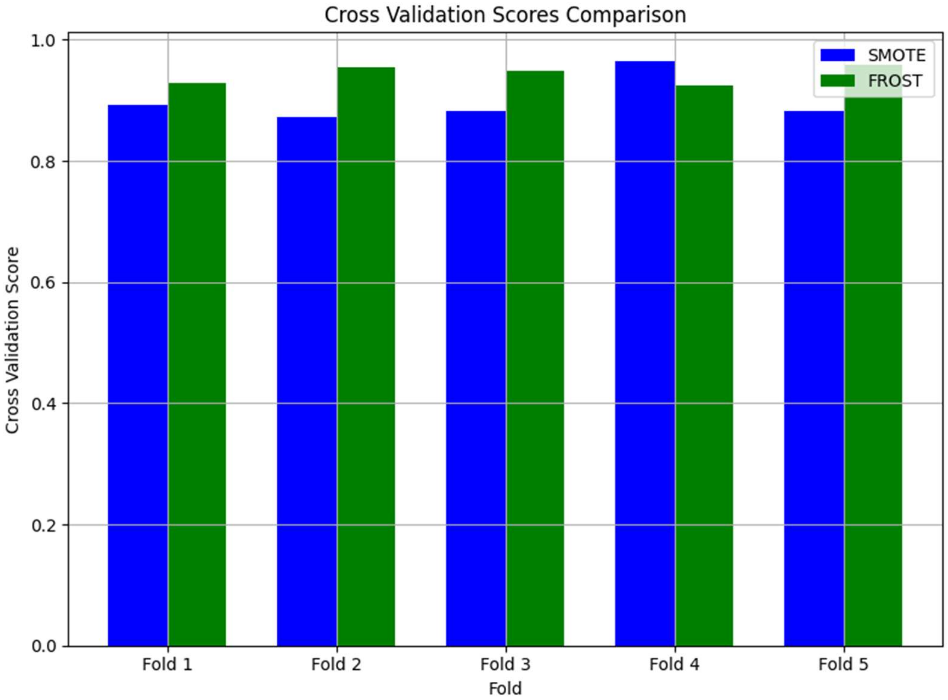


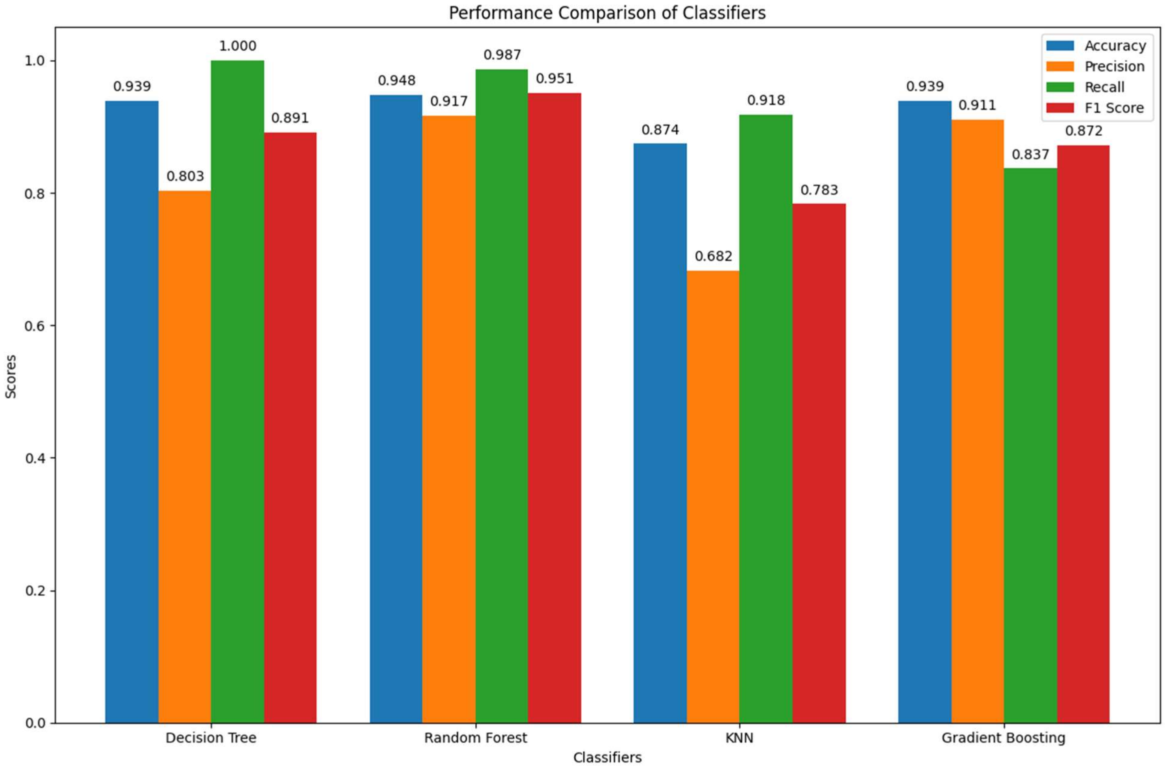
Figure 3. Percentage of Clusters generated using object weight positional value for a term/field.

```
SMOTE Cross Validation Scores: [0.89508197 0.83934426 0.8852459    0.87540984 0.92434211]
FROST Cross Validation Scores: [0.92929293 0.95454545 0.94949495 0.92424242 0.95959596]
Average SMOTE CV Score:    0.904851164797239
Average FROST CV Score:    0.9434343434343434
## HYPERPARAMETER TUNING LOGISTIC REGRESSION WITH SMOTE
Best Hyperparameters: {'C': 10.0, 'penalty': 'l1', 'solver': 'liblinear'}
Model Evaluation Metrics on Resampled Data- SMOTE:
Accuracy: 0.6312335958005249
Precision: 0.6269035532994924
Recall: 0.6482939632545932
F1 Score: 0.6374193548387097
Confusion Matrix:
[[468 294]
[268 494]]
## HYPERPARAMETER TUNING LOGISTIC REGRESSION WITH FROST
Best Hyperparameters: {'C': 1.0, 'penalty': 'l1', 'solver': 'liblinear'}
Model Evaluation Metrics on Resampled Data- FROST:
Accuracy: 0.7696969696969697
Precision: 0.5
Recall: 0.02631578947368421
F1 Score: 0.05
Confusion Matrix:
[[756     6]
[222     6]]
```

S .No.	# Evaluating with SMOTE for different Classifiers	# Evaluating with FROST for different Classifiers

1. Results for Decision Tree Classifier:	Accuracy: 0.9114754098360656 Precision: 0.9012345679012346 Recall: 0.9299363057324841 F1 Score: 0.9153605015673981 Confusion Matrix: [[132 16] [11 146]]	Accuracy: 0.9393939393939394 Precision: 0.8032786885245902 Recall: 1.0 F1 Score: 0.8909090909090909 Confusion Matrix: [[137 12] [0 49]]
Results for Random Forest Classifier:	Accuracy: 0.9475409836065574 Precision: 0.9171597633136095 Recall: 0.9872611464968153 F1 Score: 0.9509202453987731 Confusion Matrix: [[134 14] [2 155]]	Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1 Score: 1.0 Confusion Matrix: [[149 0] [0 49]]
Results for K-Nearest Neighbors (KNN):	Accuracy: 0.8459016393442623 Precision: 0.7696078431372549 Recall: 1.0 F1 Score: 0.8698060941828256 Confusion Matrix: [[101 47] [0 157]]	Accuracy: 0.8737373737373737 Precision: 0.6818181818181818 Recall: 0.9183673469387755 F1 Score: 0.782608695652174 Confusion Matrix: [[128 21] [4 45]]
Results for Gradient Boosting Classifier:	Accuracy: 0.9245901639344263 Precision: 0.8988095238095238 Recall: 0.9617834394904459 F1 Score: 0.9292307692307693 Confusion Matrix: [[131 17] [6 151]]	Accuracy: 0.9393939393939394 Precision: 0.9111111111111111 Recall: 0.8367346938775511 F1 Score: 0.8723404255319148 Confusion Matrix: [[145 4] [8 41]]

doi:10.20944/preprints202404.1461.v1



HyperParameterTune the RandomForest Classifier

Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}

Best Model Evaluation Metrics:

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

Confusion Matrix:

```
[[149   0]
 [  0  49]]
```

Note: Results will be generated using python.

5. Conclusions

This study investigates fraud detection in online transactions using a dataset containing transaction details, customer information, and merchant data. The dataset undergoes thorough analysis and feature engineering utilizing SMOTE (Synthetic Minority Over-Sampling Technique) and FROST (Feature-Space Oversampling Technique) techniques to address the imbalanced nature of the data. Cross-validation experiments reveal that FROST outperforms SMOTE for the given dataset, particularly in improving the minority class representation.

A Random Forest classifier is employed for the classification task, taking advantage of its capacity to handle intricate datasets and detect non-linear correlations. Hyper-parameter tuning is applied to optimize the Random Forest model's performance. The outcomes show that the Random Forest classifier obtains 100% accuracy on the dataset, which is corroborated by other evaluation metrics that surpass 95%, including precision, recall, and F1-score.

Overall, this study showcases the effectiveness of FROST in enhancing fraud detection performance in online transactions, and highlights the robustness of the Random Forest classifier when coupled with appropriate oversampling techniques and hyper-parameter tuning.

Author Contributions: Conceptualization: M.D.V.P, S.T.; Methodology: M.D.V.P, S.T.; Software: M.D.V.P.; Validation: M.D.V.P.; Formal analysis: S.T.; Investigation: M.D.V.P, S.T.; Resources: M.D.V.P, S.T.; Data Curation: M.D.V.P.; Writing (original draft): M.D.V.P.; Writing (review & editing): M.D.V.P.;Visualization: M.D.V.P, S.T.;Supervision, S.T.; Project administration: S.T.; Funding Acquisition: M.D.V.P, S.T.

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

References

1. Elshaar, Sulaf, and Samira Sadaoui. "Semi-supervised classification of fraud data in commercial auctions." Applied Artificial Intelligence 34.1 (2020): 47-63.
2. Hasani, Navid, et al. "Artificial intelligence in medical imaging and its impact on the rare disease community: threats, challenges and opportunities." PET clinics 17.1 (2022): 13-29.
3. Shaikh, Sarang, et al. "Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models." Applied Sciences 11.2 (2021): 869.
4. Hasib, Khan Md, et al. "A survey of methods for managing the classification and solution of data imbalance problem." arXiv preprint arXiv:2012.11870 (2020).
5. Hoyos-Osorio, J., et al. "Relevant information undersampling to support imbalanced data classification." Neurocomputing 436 (2021): 136-146.
6. Mienye, IbomoiyeDomor, and Yanxia Sun. "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data." Informatics in Medicine Unlocked 25 (2021): 100690.
7. Feng, Fang, et al. "Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification." IEEE Access 8 (2020): 69979-69996.
8. Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems.
9. Liang, X. W., Jiang, A. P., Li, T., Xue, Y. Y., & Wang, G. T. (2020). LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM. Knowledge-Based Systems, 196, 105845.
10. Raghuwanshi, B. S., & Shukla, S. (2020). SMOTE based class-specific extreme learning machine for imbalanced learning. Knowledge-Based Systems, 187, 104814.
11. Xu, T., Coco, G., & Neale, M. (2020). A predictive model of recreational water quality based on adaptive synthetic sampling algorithms and machine learning. Water research, 177, 115788.
12. Hu, Z., Wang, L., Qi, L., Li, Y., & Yang, W. (2020). A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network. IEEE Access, 8, 195741-195751.
13. Balaram, A., &Vasundra, S. (2022). Prediction of software fault-prone classes using ensemble random forest with adaptive synthetic sampling algorithm. Automated Software Engineering, 29(1), 6.
14. Al Majzoub, H., Elgedawy, I., Akaydın, Ö., &KöseUlukök, M. (2020). HCAB-SMOTE: A hybrid clustered affinitive borderline SMOTE approach for imbalanced data binary classification. Arabian Journal for Science and Engineering, 45(4), 3205-3222.
15. Smiti, S., &Soui, M. (2020). Bankruptcy prediction using deep learning approach based on borderline SMOTE. Information Systems Frontiers, 22(5), 1067-1083.
16. Sun, Y., Que, H., Cai, Q., Zhao, J., Li, J., Kong, Z., & Wang, S. (2022). Borderline smote algorithm and feature selection-based network anomalies detection strategy. Energies, 15(13), 4751.
17. Chen, Y., Chang, R., &Guo, J. (2021). Effects of data augmentation method borderline-SMOTE on emotion recognition of EEG signals based on convolutional neural network. IEEE Access, 9, 47491-47502.
18. Ayu, P. D. W., Pradipta, G. A., Huizen, R. R., &Artana, I. (2024). Combining CNN Feature Extractors and Oversampling Safe Level SMOTE to Enhance Amniotic Fluid Ultrasound Image Classification. International Journal of Intelligent Engineering & Systems, 17(1).
19. Srinilta, C., &Kanharattanachai, S. (2021, April). Application of natural neighbor-based algorithm on oversampling smote algorithms. In 2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST) (pp. 217-220). IEEE.
20. Swana, E. F., Doorsamy, W., &Bokoro, P. (2022). Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset. Sensors, 22(9), 3246.
21. Ning, Q., Zhao, X., & Ma, Z. (2021). A novel method for Identification of Glutarylation sites combining Borderline-SMOTE with Tomek links technique in imbalanced data. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(5), 2632-2641.
22. Anggrawan, A. Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link. JOIV: International Journal on Informatics Visualization.
23. Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

24. He, Haibo, et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). Ieee, 2008.
25. Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning." International conference on intelligent computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
26. Bunkhumpornpat, Chumphol, Krung Sinapiromsaran, and ChidchanokLursinsap. "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem." Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13. Springer Berlin Heidelberg, 2009.
27. Zeng, Min, et al. "Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data." 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS). IEEE, 2016.

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.