Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 18 (2023) pp. 135-146
© MIR Labs, www.mirlabs.net/jias/index.html

Vehicle Insurance Fraud Detection Based on Hybrid Approach for Data Augmentation

Zainab Saad Rubaidi [1,2*], Boulbaba Ben Ammar [1] and Mohamed Ben Aouicha[1]

[1]Data Engineering and Semantics Research Unit, Faculty of Sciences, University of Sfax, Sfax, Tunisia.

[2] College of Agriculture, Al-Muthanna University, Samawah, Iraq,
*Corresponding author(s). E-mail(s): Zainabalkalidy@mu.edu.iq;
Contributing authors: Boulbaba.BenAmmar@fss.usf.tn;
Mohamed.BenAouicha@fss.usf.tn;

Abstract: Fraud can take on various forms, including financial fraud, identity theft, and insurance fraud, among others. With the growing use of technology, fraudulent activities have become more sophisticated, making it difficult for organizations to detect and prevent them. One major challenge in the insurance industry is vehicle insurance fraud, which leads to increased expenses and a loss of trust. Using machine learning techniques has gained prominence as an efficient approach for detecting fraud. This paper aims to test the performance of various supervised machine learning models using different data resampling techniques (undersampling and oversampling) for vehicle insurance fraud detection. This study compares the performance of NearMiss, SMOTE, and our proposal hybrid data augmentation approach for data resampling. The preprocessing steps used in the methodology include dropping irrelevant features, filling missing values, encoding features with dummy variables, and selecting features using a correlation approach. The testing results indicated that of Random Forest (RF) model performed best using our proposal hybrid data augmentation approach achieving the highest F1-score of 0.975 and accuracy of 0.975 in fraud detection.

Keywords: Fraud, Machine Learning, Hybrid Data Augmentation, Data oversampling.

I. Introduction

Fraud detection plays a crucial role in multiple industries, as it helps prevent financial losses and sustain the integrity of systems and processes [1]. The adoption of deep learning (DL) and machine learning (ML) approaches for fraud identification has seen a marked rise in recent years, due to the ability of these methods to process large and complex datasets, and identify patterns and anomalies that may suggest fraudulent behavior [2].

Studies on data mining-based fraud detection have demonstrated the usefulness of the ML methods, like decision trees and random forests, in building predictive models Studies on data mining-based fraud detection have demonstrated the usefulness of the ML methods, like decision trees and random forests, in building predictive models that flag potential fraudulent activities based on historical data [3]. Additionally, the DL methods, for example, neural networks, are widely utilized in fraudulent activities uncovering due to their capability of detecting complex relationships in data [4]. The deep learning models' capacity to learn from data and uncover complex relationships has made them particularly suitable for fraud detection, as they are able to accurately identify patterns that may suggest fraudulent behavior [5]. Vehicle insurance fraud presents a significant challenge in the insurance industry, leading to increased costs and decreased trust. To overcome this issue, the application of machine learning techniques has gained recognition as an effective method for detecting fraud. Machine learning algorithms can analyze large datasets, identify patterns and anomalies, and predict potential fraud cases [2]. Another research has demonstrated that ML approaches, such as decision trees and random forests, can be utilized to construct predictive models for detecting vehicle insurance fraud [6].

These models can determine potential fraud cases based on historical data, including policy information, claims history, and demographic data [3]. Furthermore, DL techniques, such as neural networks, are also being applied for fraud detection as they have the capability to detect intricate relationships in the data [4]. The use of machine learning techniques has been established as an effective solution for vehicle insurance fraud detection [4]. By analyzing large and complex datasets, machine learning algorithms can precisely detect patterns and anomalies that may signify fraudulent behavior, which contributes to preventing financial losses and improving trust in the insurance industry [7]. In general, most exiting fraud detection datasets have imbalance class problem. However, this study attempts to give answers for our two research questions that are listed as follows:

- What is the best ML technique for vehicle insurance fraud analysis and detecting?
- What is best resampling technique for fraud dataset balancing?

The aim of this study is to match the testing of different supervised machine learning algorithms, including Random Forest, XGBoost, and Adaboost, SVC, Logistic Regression and KNN with different data resampling techniques such as data undersampling, data oversampling and a hybrid data augmentation which is our proposal data resampling approach based on combination of various data oversampling techniques, for the detection of vehicle insurance fraud. This is to provide answers to our research questions. This article extends upon the findings presented in the conference paper [31].

II. Related works

This section highlights on some related research that had proposed solutions and techniques for fraud detection in previous years. It can be divided into two subsections such as follows:

A. Review for fraud detection problems

In recent years, fraud detection has gained significant attention due to the growing volume of financial transactions conducted online and the widespread adoption of electronic payment systems. Fraud detection involves identifying any illegal activities or attempts at deception through the examination of data patterns.

Based deep learning techniques, The authors in [8] investigated a detection of credit card counterfeits using three different algorithms: Convolutional Neural Network (CNN), integrating CNN with Gated Recurrent Units (GRU), and Adaptive Boosting (AdaBoost). The study implements an oversampling technique, Synthetic Minority Oversampling Technique (SMOTE), to solve the high unbalance data class problem in the dataset. The performance evaluation metrics show that the CNN algorithm beats the other techniques, achieving high accuracy, precision, AUC-ROC and recall rates.

Based on credit card fraud dataset used, four ML based models (artificial neural networks, stacked ensemble, gradient boosting machine, and random forest) were trained on various sampling techniques, including random undersampling, SMOTE, density-based SMOTE, and SMOTE + ENN. The results indicated that SMOTE-based sampling methods produced encouraging outcomes, with the best recall score achieved through the SMOTE method applied to the random forest classifier. As a result, the authors deemed the SMOTE technique to be the most preferred. [9].

Saputra (2019) suggested that the utilization of the ML methods is to be implemented for fraud prevention in e-commerce. The aim is to examine the optimal machine learning algorithm; Decision Tree, Naive Bayes, Random Forest (RF), and Neural Network are the algorithms were used. The data to be utilized is still imbalanced, therefore the Synthetic Minority Over-sampling Technique (SMOTE) will be employed to balance the data. Evaluation results using a confusion matrix indicate that the highest accuracy was achieved by the Neural Network at 96%, followed by Random Forest at 95%, Naive Bayes at 95%, and Decision Tree at 91%. [10].

Rubaidi et al. (2022) developed a framework to tackle imbalance datasets for credit fraud detection. The authors tested various resampling techniques including data oversampling and undersampling on a big size unbalanced dataset collected from the Kaggle website. The dataset was used to detect fraud in a Tunisian company for electricity and gas consumption. The performance of the framework was evaluated using Logistic Regression, Naive Bayes, Random Forest, and XGBoost machine learning classifiers. The results were measured using precision, recall, F1-score, and accuracy metrics. The findings indicated that the Random Forest model performed the best, achieving 89% accuracy with NearMiss undersampling and 99% accuracy with random oversampling [11].

Fraud analysis in Healthcare Insurance domain research was presented by [12],[31] the authors have been proposed a fraud detection method for healthcare insurance claims that utilizes an improved support vector machine (SVM) algorithm with oversampled SMOTE and particle swarm optimization. The study finds that the results for fraud detection have been improved with using SMOTE and SVM classifier.

The authors in [13] introduced new approach to balancing fraud detection datasets, which are typically highly unbalanced, by using a Generative Adversarial Network (GAN) to generate synthetic fraudulent transaction data. The authors claimed that their method improved precision and F1-score compared to traditional oversampling techniques like SMOTE, ADASYN, and random oversampling, and reduces the number of false positives. The effectiveness of the proposed method was evaluated through an ablation study.

Botchey et al. (2020) focused on mobile money fraud prediction using three machine learning algorithms: Support Vector Machines (SVM), Gradient Boosted Decision Trees (GBDT), and Naive Bayes (NB). The authors perform a cross-case investigation to evaluate the effectiveness of these algorithms with in detecting mobile money fraud. They used SMOTETomek resampling technique for dataset balancing purpose and compared the results of the three algorithms using various evaluation metrics such as accuracy, precision, recall, and F1-score. Their paper provided insight into the strengths and weaknesses of the different algorithms, and their proposed results of the exploration were expected to inform the development of better fraud prediction models for mobile money transactions [14].

B. Review on data oversampling techniques

Data oversampling is a method for mitigating the imbalance between classes in machine learning datasets. The imbalance occurs when there are fewer samples in the minority class compared to the majority class. The imbalance class problem can lead to develop biased algorithms that struggle to accurately classify samples in the minority class. To resolve this problem, data oversampling techniques are utilized to artificially enhance the count of examples in the minority class. Replication is one type of data oversampling approach. This implicates duplicating samples in the minority class to attain a balanced class distribution. Synthetic Minority Over-sampling technique (SMOTE): This involves generating synthetic samples in the minority class by interpolating between existing minority class samples [15].

Adaptive Synthetic (ADASYN) oversampling: This involves creating artificial samples in the minority class with higher density in regions of the feature space where the minority class is under-represented [16].

Borderline-SMOTE: This involves generating synthetic samples in the minority class near the decision boundary among minority and majority classes [17].

Cost-Sensitive Oversampling: This involves weighting the samples in the minority class based on the cost of misclassifying them to make balance between class distribution [18].

A hybrid data oversampling approach is a combination of two or more data oversampling techniques aimed at creating a more effective and efficient solution for imbalanced data sets. Imbalanced data sets occur when the dissemination of classes in a dataset is unequal, which can result in biased machine learning models. Hybrid data oversampling approaches aim to overcome the limitations of individual oversampling techniques and improve the evaluation of the ML algorithms.

One common hybrid approach is the combination of random oversampling and SMOTE. Random oversampling duplicates existing minority samples to balance the class distribution, while SMOTE creates novel artificial samples by incorporating among existing minority samples. By combining these two techniques, the hybrid approach can address the overfitting problem associated with random oversampling while still improving the balance of the class distribution [9].

Another popular hybrid data resampling approach can be created by the combination of random oversampling and cost-sensitive learning. Cost-sensitive learning is a machine learning technique that assigns different misclassification costs to different classes, which allows the algorithm to account for the imbalance in the class distribution. By combining random oversampling with cost-sensitive learning, the hybrid approach can improve the balance of the class distribution while still considering the cost of misclassifying different classes [18].

Several studies have reported improved performance using hybrid data oversampling approaches compared to individual oversampling techniques. For example, a study by He and Garcia (2009) found that the grouping of random oversampling and SMOTE improved the accuracy of a decision tree classifier by 6% compared to using random oversampling alone [19]. Similarly, Le et al. (2017) found that the integrating of random oversampling and cost-sensitive learning improved the precision and recall of a SVM classifier compared to using random oversampling alone [20].

Hybrid data oversampling approaches offer a promising solution for handling imbalance data class problem while training machine learning algorithm. By combining two or more oversampling techniques, hybrid approaches can address the limitations of individual techniques and improve the performance of the ML algorithms. Further research is needed to determine the best combination of oversampling techniques for specific applications and to validate the results obtained from these hybrid approaches.

Chen et al. (2021) developed new approach for addressing class imbalance in datasets is presented. The method, called Hybrid Sampling Method based on Data Partition (HSDP), partitions all data examples into different regions and selectively removes noise minority samples and oversamples boundary minority samples using weighted oversampling that considers the creating of artificial examples within the same cluster based on oversampling seed .the authors compared the performance of their proposed hybrid sampling method (HSDP) against the other techniques, comparative experiments were performed, including SMOTE, ADASYN, and Borderline-SMOTE [21].

Another hybrid approach, called Adaptive Synthetic Sampling (ADASYN), has been proposed by [16]. ADASYN is a combination of random oversampling and SMOTE, with the added feature of adaptively adjusting the oversampling rate based on the difficulty of each minority class sample. The idea is that the oversampling rate for samples with similar surrounding samples should be lower than for samples in isolated regions. The results of several experiments showed that ADASYN outperformed both random oversampling and SMOTE in terms of classification accuracy.

Another hybrid approach is "SMOTE-ENN" (SMOTE with edited nearest neighbors), which was proposed by Kovács. (2019), the author combined the SMOTE oversampling method with the edited nearest neighbor (ENN) technique to address the problem of imbalanced data. SMOTE creates synthetic samples by interpolating between minority class samples, while ENN removes examples that are not representative of the minority class. By combining these two techniques, the authors were able to expand the enactment of the model and reduce over-generalization. The results of experiments on several datasets showed that the SMOTE-ENN approach outperformed other oversampling techniques such as random oversampling and SMOTE itself [22].

A hybrid method referred to as "SMOTE-Tomek links" was presented by Batista et al. (2004). This approach blends the SMOTE oversampling with the Tomek links method to handle imbalanced data classes. Tomek links are group samples from dissimilar classes that are located neighboring to each other and removing them can enhance the model's testing results. The combination of SMOTE and Tomek links enhances the performance of predictive machine learning model and reduces over-generalization. Their proposed results on various datasets revealed that the SMOTE-Tomek links method outclasses other oversampling techniques, such as random oversampling and SMOTE by itself [23].

Hybrid data oversampling methods hold great potential for addressing the issue of imbalanced data in machine learning. By blending two or more oversampling techniques, these methods can overcome the drawbacks of singular methods and enhance machine learning algorithm performance. Further studies are necessary to identify the optimal combination of oversampling techniques for specific situations and to confirm the results achieved from these hybrid methods.

III. Hybrid data augmentation approach(our proposal)

This subsection demonstrates the structure details of our proposal hybrid approach for data augmentation. Figure 1 depicts a framework of our approach which is consisting of steps such as dividing dataset classes, majority class samples portioning, data concatenation and balanced dataset. More details on these steps can be described in next section. In the initial step, we divided the dataset labels, in this step, separating the data points presented in the dataset into different classes or categories into two dataframes, based on their class label.

This is typically done in order to find the minority and majority classes, where the goal is to divide the majority class samples into equal k-folds. Further, we portioned the majority class samples into five k-folds which is similar as cross-validation technique used in machine learning.

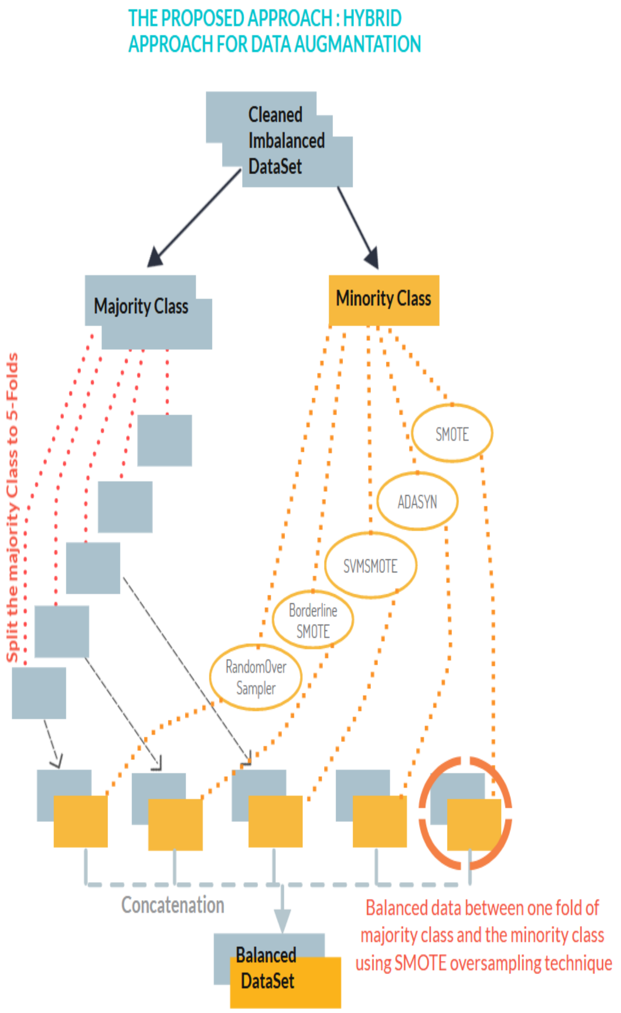

THE PROPOSED APPROACH : HYBRID
APPROACH FOR DATA AUGMANTATION
Cleaned
Imbalanced
DataSet
Majority Class
Minority Class
Split the majority Class to 5-Folds
SMOTE
ADASYN
SVMSMOTE
Borderline
SMOTE
RandomOver
Sampler
Concatenation
Balanced
DataSet
Balanced data between one fold of
majority class and the minority class
using SMOTE oversampling technique

Figure 1. Structure of our proposed approach

Moreover, we used five data oversampling techniques such random, smote, borderline smote, adasyn and svmsmote where each takes one-fold of the majority samples and creates synthetic samples of minority class samples till equal to the samples presented in each fold. Then automated concatenation process is used to merge all folds into one dataframe and finally, balanced dataset is generated at the end of this process and splitted into training and testing various machine learning algorithm such as Random Forest, XGboost, Adaboost, SVM, KNN and Logistic Regression for fraud classification task.

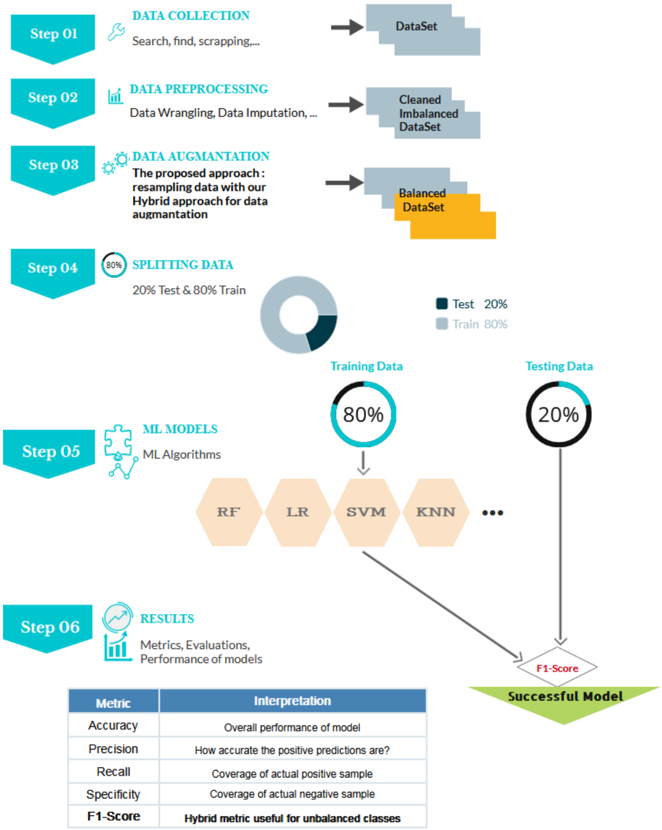
IV. Methodology

In this section, a detailed methodology is presented for detecting vehicle fraud. The methodology includes a series of important steps that must be followed in order to achieve this goal. The first step is to collect the necessary data. This data is then preprocessed to ensure that it is in the proper format for analysis.

After the preprocessing step, the data is splitted into two parts: a training set and a testing set. The training set is utilized to train the machine learning algorithms, whereas the testing set is used to evaluate the performance of the used ML algorithms.

The next step is to apply various classification techniques to the training set in order to develop a predictive model that can accurately detect vehicle fraud. The results of the ML model are then evaluated using appropriate metrics, such as accuracy, precision, recall, and F1 score. The results are then presented, providing insight into the effectiveness of the proposed methodology for detecting vehicle fraud.

For a visual representation of the methodology, refer to Figure 2. This figure provides an overview of the different components that make up the proposed methodology for detecting vehicle insurance fraud, highlighting the steps involved and the flow of the process.



Metric	Interpretation
Accuracy	Overall performance of model
Precision	How accurate the positive predictions are?
Recall	Coverage of actual positive sample
Specificity	Coverage of actual negative sample
F1-Score	**Hybrid metric useful for unbalanced classes**

Figure 2. The framework of the proposed methodology

The components of a framework can be discussed as follows:

1) *Step 01: Data collection*

The act of obtaining, searching, scrapping, documenting, and preserving data is known as data collection. This information is frequently collected with the intention of later examination to reach knowledgeable decisions or arrive at conclusions. In the realm of machine learning, dataset collection is a crucial step in creating a model. The model's accuracy and effectiveness will be significantly influenced by the quantity and quality of the collected data. After the dataset is collected, it is processed and cleaned, if necessary, before it is utilized to train and evaluate machine learning algorithms.

2) Step 02: Data preprocessing and feature selection

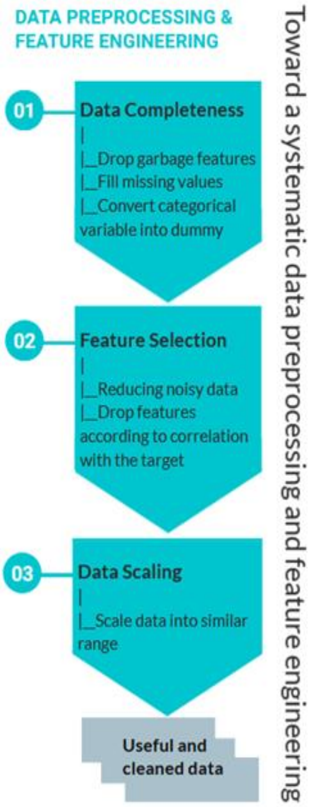

DATA PREPROCESSING &
FEATURE ENGINEERING
Toward a systematic data preprocessing and feature engineering
01
Data Completeness
|
|_Drop garbage features
|_Fill missing values
|_Convert categorical variable into dummy
02
Feature Selection
|
|_Reducing noisy data
|_Drop features according to correlation with the target
03
Data Scaling
|
|_Scale data into similar range
Useful and cleaned data

Figure 3: Workflow of data preprocessing steps.

The process of preparing data for analysis is referred to as data preprocessing, and it is considered a vital stage in both data mining and machine learning. Figure 3 shows Workflow of data preprocessing step. To develop a reliable model, the data must undergo various preprocessing techniques to reach a format that is appropriate for analysis.

In the context of machine learning, data completeness pertains to the degree to which a dataset encompasses all the significant information necessary to attain the expected results. Essentially, it signifies the fraction of data that lacks missing values within the dataset. It is one type of data preprocessing steps, which is also referred to as data munging, involves cleaning, transforming, and organizing data.

This step has been included tasks such as eliminating and replacing missing or inaccurate values, addressing duplicates, transforming categorical data features into dummy variables. Feature selection is also preprocessing steps aims to select important features from the dataset.

In this step, we used correlation approach as features selection which was applied to match the correlation between features and eliminate one of two features that have a correlation higher than 0.9. The next data preprocessing step is featuring scaling, which aims to scale the dataset features values in the same range.

3) Step 03: Data augmentation

Data augmentation through oversampling is a strategy for enhancing the size of a training dataset in machine learning. The objective of this technique is to equalize the class distribution of the data by duplicating examples from the underrepresented class, known as the minority class, until it has the same number of samples as the majority class.

This duplication process is referred to as oversampling. By exposing the model to more examples of the minority class, it can learn the patterns and differences between classes more effectively. This strategy can be useful in situations where the original dataset has an imbalanced class distribution, resulting in a model that is biased towards the majority class. The use of oversampling can help improve the model's accuracy, especially when the cost of false negatives (not detecting the minority class) is high. Figure 4 demonstrates the distribution of the used dataset after resampling.

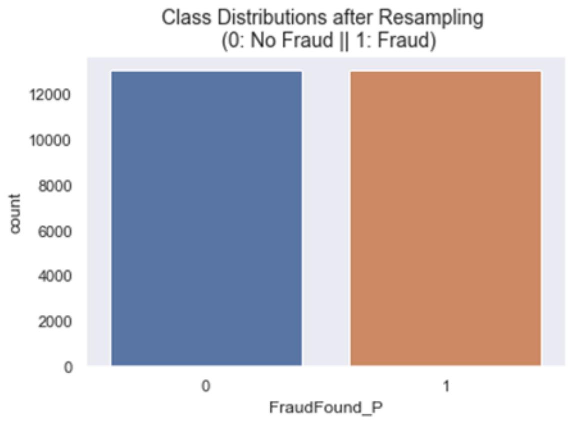


Figure 4: A distribution of the dataset after resampling

4) Step 04: Data splitting

Dataset splitting is a process of dividing an evaluated data into train and test sets. In the proposed methodology, the used dataset was split into 80% training and 20%. Testing process refers to a common practice while evaluating the ML models. The training set is adopted to train the machine learning model, while the testing set is used to evaluate the performance of the trained model. This data split is a rough guideline and can vary depending on the size of the dataset and the goals of the analysis. The training set is typically larger than the testing set as it is used to learn the patterns in the data, while the testing set is used to assess the model's performance and check for overfitting. Overfitting occurs when the model is too complex and fits the training data too closely, leading to poor generalization to new, unobserved data.

5) Step 05: Machine learning models

Classification in machine learning involves training a model to categorize inputs based on formerly categorized data, with the objective of accurately predicting the category of new, unseen inputs. In this step, we implement various supervised machines learning models for classification the dataset instances into fraud and non-fraud. These techniques were Random Forest

(RF), Adaboost, and Xgboost, SVM, Logistic Regression and KNN.

5.1) Random Forest model

RF is classifier is a type of ensemble learning method for classification problems in machine learning. It is an extension of decision tree algorithm, where several decision trees are united to create a forest and the ultimate estimation is prepared by compelling the average of all the trees. This algorithm can be worked by creating several decision trees from arbitrarily nominated subdivision of the training data and features. Every decision tree part is learnined on a different subset of the data and features, and the final prediction is made by combining the outputs of all the trees. This combination results in a more robust model with reduced overfitting, compared to a single decision tree [25].

5.2) *XGBoost model*

XGBoost is a great ML method that is widely utilized for solving various classification and regression predictive problems. It is an optimized implementation of gradient boosting, which is a popular collaborative learning method that syndicates several simple models, such as decision trees, to produce a more complex model that provides better results. The key point of XGBoost is its capability to handle large datasets and perform computations efficiently, making it a popular choice for data science and machine learning competitions [26]. It also provides several advanced features, such as regularization to prevent overfitting, parallel processing to speed up computations, and the handling of missing values, making it a versatile and flexible algorithm.

5.3) AdaBoost model

AdaBoost (Adaptive Boosting) is a collaborative learning technique adopted for binary and multi classification problems. It works by combining several weak classifiers to form a strong classifier that can accurately predict the target class [27]. In AdaBoost, each weak classifier is trained on the entire dataset, but the importance of each instance in the training set is adjusted based on the performance of the previous weak classifier. This allows AdaBoost to focus on the samples that are difficult to classify, and improve the overall accuracy of the model. In this study, the Adaboost classifier was trained and tested with 20 estimators.

5.4) KNN model

The K-Nearest Neighbors (KNN) algorithm is a popular tool in supervised machine learning, which is used to tackle both classification and regression tasks. This method operates under the premise of supervised learning, where the target variable is known, and the algorithm is trained on a labeled dataset. The KNN algorithm functions by utilizing the information from the training data to make predictions for new, unseen instances. To make a prediction, the algorithm first identifies the K nearest neighbors in the training data, based on a selected similarity metric such as Euclidean distance. Then, it takes a majority vote among those K neighbors to determine the label for the new data point [28].

5.5) SVM model

Support Vector Machines (SVM) is a supervised ML technique employed for either classification or regression tasks. The objective of SVM is to identify the optimal line or hyperplane that parts the data points into distinct classes. This line or hyperplane, which provides the greatest margin or distance between the nearest data points of each class, is devoted to as the maximum margin classifier. The data points located closest to the margin are referred to as support vectors and play a crucial role in determining the position of the hyperplane. The optimization challenge in SVM involves finding the hyperplane with the determined margin that separates the classes. This problem is known as the primal problem, which can be transformed into a dual problem, which is then solved to determine the support vectors [29].

5.6) Logistic Regression *model*

Logistic Regression is a statistical approach that allows the analysis of datasets where there are one or more independent variables that influence an outcome. It is specifically used in binary classification problems, where the target variable has two possible outcomes, such as yes/no, true/false, and so on. The main principal concept of logistic regression is to find the connection between the independent variables and the dependent variable by modeling the probability of the relianted variable's occurrence as a function of the independent variables. The resultant model is then employed to make predictions about the dependent variable based on the values of the independent variables. The logistic regression model is a form of generalized linear model, which utilizes a logistic function to model the probability of the dependent variable. The logistic function is used to map the predicted probabilities to a value between 0 and 1. The coefficients of the independent variables in the logistic regression model are calculated using the maximum likelihood estimation method [30].

6) Step 06: Evaluation metrics

Performance evaluation was conducted using comprehensive evaluation measures, comprising precision, accuracy, recall, and F1-score. Metrics of performance measurement were analyzed to inspect the results of the proposed machine learning algorithms for fraud and non-fraud client discrimination. During the testing phase of each algorithm, a confusion matrix was utilized to capture various results, containing False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN). A False Positive (FP) is a non-fraud client who was incorrectly classified as fraud, a True Negative (TN) is a fraud client who was accurately categorized as fraud, and a False Negative (FN) is a non-fraud client who was mistakenly identified as fraud. The first metric is the test accuracy, which is the proportion of the total number of correct predictions. The Equation 1 represents the formula for quantifying the accuracy.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

The second metric is the F1-score, which represents the harmonic mean of the values of Precision (Positive Predictive Value) and Recall (Sensitivity) for a classification problem. The calculation of this metric is illustrated by the Equation 2).

$$Precision = TP / (TP + FP)$$

$$Sensitivity = Recall = TP / (TP + FN)$$

$$Specificity = TN / (TN + FP)$$

$$F1-score = 2 * Precision * Recall / (Precision + Recall) \quad (2) \quad (eq.2)$$

The last metric is the AUC, which signifies the Area Under the ROC Curve. The ROC curve has the True Positive Rate, called Sensitivity, on the y-axis against the False Positive Rate, which is calculated as (1-Specificity), on the x-axis, at various cut-off thresholds of a binary classifier.

V. Case study: Vehicle insurance fraud detection.

This section presents the details implementation steps of our proposed methodology for vehicle insurance fraud detection:

1) Vehicle insurance fraud dataset

The data set used for conducting our experimental work has been collected from kaggle platform [24]. It consists of vehicle insurance claim information obtained from the Angoss Knowledge Seeker Software, commonly referred to as "carclaims.txt". The data encompasses 15420 claims from January 1994 to December 1996, with 32 predictor variables and one target variable, which indicate whether a claim is "Fraud" or "No Fraud". The data set contains 14,497 genuine (non-fraud) claims (94%) and 923 fraud instances (6%), with an average of 430 claims per month. Additionally, the data set has 6 numeric features and 25 alphanumeric features, with a total of 33 features per instance, including:

- Personal information of the insured (such as age, gender, marital status, etc.)
- Insurance contract details (such as policy type, vehicle category, deductible insurance payments, number of supplements, agent type, insurance coverage, etc.)
- Circumstances surrounding the accident (such as the date and location of the accident, policy report filed, presence of witnesses, fault liability, etc.)
- Other information related to the insured (such as the number of cars, previous claims, driver rating, etc.)
- The target feature indicates whether fraud was found (yes or no).

2) Data preprocessing

Data preprocessing is considered a crucial step in both data mining and machine learning. This is since large datasets often contain noisy, incomplete, inconsistent, or redundant data. To produce a robust model, the data must undergo a series of preprocessing procedures to reach a suitable format. Only after preprocessing can an appropriate training and testing dataset be obtained. In data preprocessing, we performed various steps on the used vehicle insurance fraud dataset such as dropping garbage features, filling missing values, converting categorical features into dummy variables. Further, we used Pearson correlation approach as feature selection to select important features by comparing the correlation between features and remove one of two features that have a correlation higher than 0.9. Preprocessing is a necessary step for any dataset before it can be utilized. This holds true for the vehicle insurance fraud detection dataset, which was also analyzed and required preprocessing. As the dataset has 33 features from that there are 25 alphanumeric features have been converted to numeric format. Three features such as PolicyNumber', 'RepNumber', 'Age' have been dropped due to their irrelevant to other features. To speed up the implementation of models, we used a StandardScaler to transform dataset features values between 1 and -1. A equation for Standard scalar is given as follows.

$$z = (x - u) / s \quad (3)$$

The mean and standard deviation of the training values are signified by u and s, correspondingly.

3) Data splitting

Dividing data into training and testing sets is a common practice in machine and deep learning. This helps evaluate both hyper-parameter tuning and generalization performance of the model. The used dataset was split into 80% training, 20% testing set. Table 1 shows the data splitting before and after resampling process.

Table 1: Data splitting

Approach	Total dataset samples	Training set	Testing set
Hybrid data augmentation (Our proposal)	28,994	23,195	5799
Without resampling (Original data)	15,420	12,336	3084
Undersampling	923	738	185
Oversampling	28,994	23,195	5799

Further, both training and testing sets were resampling using oversampled and undersampled techniques. Figure 5 shows the class distribution of dataset classes before resampling.

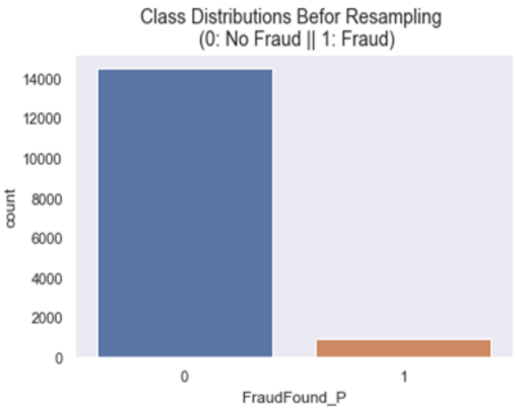


Figure 5. The distribution of the dataset classes before resampling

VI. Comparison of sampling methods

1) Evaluation of ML Algorithms based on hybrid data augmentation.

This section presents testing evaluation results obtained from the performance of used machine learning algorithms on testing set. Table 2 summarizes the testing using hybrid data augmentation approach (Our proposal).

Table 2: Testing results of the proposed ML models using hybrid data augmentation.

Model	**Specificity**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
RF	**0.954**	**0.956**	**0.995**	**0.975**	**0.975**	**0.98**
XGboost	0.757	0.798	0.964	0.874	0.861	0.86
AdaBoost	0.664	0.719	0.862	0.784	0.763	0.76
KNN	0.839	0.861	1.0	0.925	0.919	0.920
SVC	0.894	0.903	0.984	0.942	0.939	0.94
LR	0.712	0.747	0.851	0.796	0.781	0.78

2) Evaluation of ML Algorithms in original data

In the second experiment, the used machine learning algorithms were trained and tested on original dataset samples without using any resampling approach. For comparing the testing results of experimented machine learning models, the first experiments were conducted on the dataset without resampling. In this experiment each model has been trained 12,336 and tested on 3084 samples. Table 3 displays the testing results before data resampling.

Table 3: Testing results of the proposed models using original data.

Model	**Specificity**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
RF	0.994	0.423	0.059	0.104	0.939	0.53
XGboost	1.0	1.0	0.043	0.082	0.942	0.52
AdaBoost	0.994	0.210	0.021	0.039	0.936	0.51
KNN	0.988	0.179	0.037	0.062	0.932	0.51
SVC	1.0	0.0	0.0	0.0	0.940	0.50
LR	0.998	0.4	0.010	0.021	0.940	0.50

From the results presented in table 3, it appears that the AdaBoost, SVC and KNN models have relatively low precision and recall scores, while the Random Forest, XGBoost, and LR models have high precision scores but low recall scores. This indicates that these models may be good at identifying positive cases, but they might not be catching many of the actual positive cases (i.e., cases of fraud). The testing accuracy scores for all models are around 0.93 to 0.94, which indicates that they are not performing optimally in detecting fraud.

3) Evaluation of ML Algorithms based on undersampling method

From the third experiment results performed using under sampling presented in Table 4, it is observed that the results indicate that SVC model performed the best with a precision of 0.869, recall of 0.718, F1-score of 0.786, testing accuracy of 0.805, and AUC is 0.81. The results for the other techniques are lower, but still show decent performance.

Table 4: Testing results using NearMiss undersampling approach.

Model	**Specificity**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
RF	0.762	0.739	0.675	0.706	0.718	0.72
XGboost	0.767	0.754	0.713	0.733	0.740	0.74
AdaBoost	0.8	0.784	0.729	0.756	0.764	0.76
KNN	0.783	0.714	0.540	0.615	0.662	0.66
SVC	0.891	0.869	0.718	0.786	0.805	0.81
LR	0.859	0.838	0.729	0.780	0.794	0.79

4) Evaluation of ML Algorithms based on oversampling method

In this experiment which conducted based on data oversampling techniques such smote, random, etc. The total dataset samples generated by this approach were 28,994 balanced class samples splitted into 23,195 as training set and 5799 samples as testing set. Table 5 shows testing results of the proposed models using data oversampling. While comparing the performance of adopted ML models using SMOTE method, the results showed that the XGBoost algorithm consistently performs the high precision, recall, and F1-score values, as well as high testing and training accuracy values.

Table 5. Testing results of the proposed models using data oversampling techniques.

Technique	**Model**	**Specificity**	**Precision**	**Recall**	**F1- score**	**Accuracy**	**AUC**
SMOTE	RF	0.976	0.975	0.946	0.960	0.961	0.96
	XGboost	0.998	0.998	0.944	0.970	0.971	0.97
	AdaBoost	0.836	0.847	0.910	0.878	0.878	0.87
	KNN	0.680	0.757	1.0	0.862	0.840	0.84
	SVC	0.923	0.928	0.993	0.960	0.959	0.96
	LR	0.657	0.725	0.906	0.805	0.782	0.78

VII. Results and discussion

In the case of vehicle fraud insurance detection, it is important to have a high recall rate (i.e., a low false negative rate) as it is crucial to detect as many fraud cases as possible. At the same time, a high precision rate (i.e., a low false positive rate) is also important, as it ensures that only actual fraud cases are flagged and not benign cases, which could harm the reputation of the insurance company.

To determine the best model according to the best performance metric, we need to consider the specific problem and the desired trade-off between different metrics. The F1 score would be the most appropriate performance metric to evaluate the models. The F1 score is the harmonic mean of precision and recall and provides a balance between the two.

Based on the results presented in above cited four tables (see table 2 to 5), and according to the F1 score, the Random Forest model based on proposal hybrid data augmentation approach (0.975) has the best performance among all the models, followed by the XGBoost model based on the SMOTE method of oversampling (0.971). Figure 6 depicts the best confusion matrixes generated by the RF model.

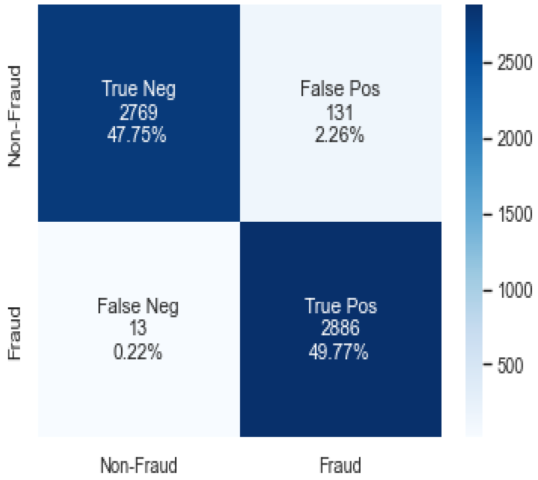


Figure 6. Confusion matrix of RF model hybrid data oversampling.

With comparing the misclassification rate obtained from the confusion matrixes of the RF model performance, it observes that the RF had less misclassification rate using hybrid data augmentation approach compared with others techniques. Figure 7 presents the AUC curves which visualize on their y-axis a true positive rate and on x-axis a false positive rate for RF model.

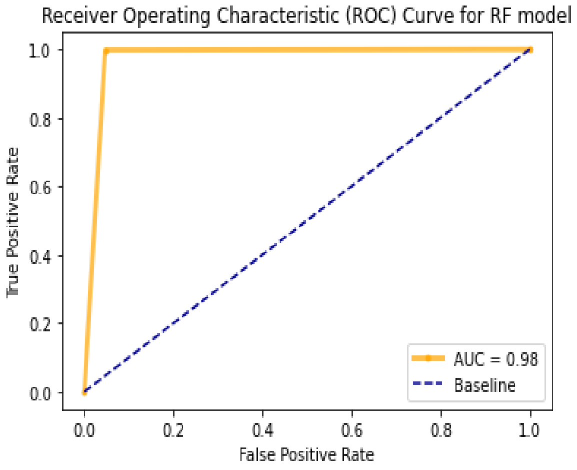


Figure 7. AUC curves for the RF using hybrid data oversampling approach.

Figure 8 shows The Precision-Recall curves of the RF mode using hybrid data augmentation approach. This curve is created by plotting the precision values against their respective recall values, using different threshold levels for the classifier. The Area under the precision-recall curve (AUPRC) is widely used as an evaluation metric for the general performance of binary classifiers as it considers both precision and recall. A higher AUPRC value indicates that the classifier is performing better in terms of balancing precision and recall.

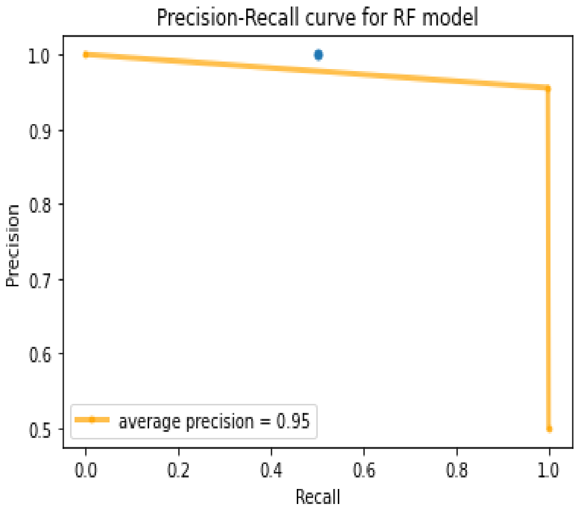


Figure 8. PR-curves of RF model using hybrid data oversampling.

Finally, it is important to note that the results be subject to on the specific dataset used, the preprocessing and feature selection techniques applied, and the parameter tuning of each model. Further experiments and tuning may improve the performance of the other models.

In conclusion, the testing results suggest that the Random Forest model with hybrid data augmentation approach is the best model for vehicle fraud insurance detection in this scenario. The high values of recall, precision, and F1 score indicate that the model is able to effectively detect fraud cases while minimizing the number of false positive cases. This makes it a suitable model for practical implementation in the field of vehicle fraud insurance detection.

VIII. Conclusions

This paper presents comparative analysis between various data resampling techniques such as undersampling using NearMiss, data over sampling using SMOTE oversampling and a proposed hybrid data augmentation (using simultaneously five oversampling methods) for data balancing purpose. According to the obtained experimental results, it can be concluded that the performance of various techniques and models for vehicle fraud analysis and detection. The models used were RF, XGBoost, AdaBoost, KNN, SVC, LR.

The precision, recall, F1-score, testing accuracy were used as evaluation metrics in testing phase. The results showed that the Random Forest model with hybrid data augmentation (our proposal) achieved the highest F1-score of 0.975 and testing accuracy of 0.975. The XGBoost model with SMOTE provided the best performance among six models. To answer our research questions given in research introduction, the best data balancing and machine learning techniques for fraud detection are hybrid data augmentation (our proposal) and RF model for classification task.

In conclusion, the choice of technique and model for vehicle insurance fraud detection depends on the desired trade-off between precision and recall, as well as the accuracy of the model. The RF model with hybrid data augmentation approaches can be considered as a promising starting point for this problem, however, the performance of other models and techniques should also be evaluated for comparison [32-38].

Acknowledgment

The authors would like to express their gratitude to the following institutions whose support and contributions were instrumental in the completion of this research paper: College of Agriculture, Al-Muthanna University, Iraq. Data Engineering and Semantics Research Unit, Faculty of Sciences, University of Sfax, Tunisia.

References

[1] Matulich, S. and Currie, D.M. eds., 2017. *Handbook of Frauds, Scams, and Swindles: Failures of Ethics in Leadership*. CRC press.

[2] Chaudhary, K., Yadav, J. and Mallick, B., 2012. A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, *45*(1), pp.39-44.

[3] Phua, C., Lee, V., Smith, K. and Gayler, R., 2010. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

[4] Abdallah, A., Maarof, M.A. and Zainal, A., 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, pp.90-113.

[5] Lebichot, B., Le Borgne, Y.A., He-Guelton, L., Oblé, F. and Bontempi, G., 2020. Deep-learning domain adaptation techniques for credit cards fraud detection. In *Recent Advances in Big Data and Deep Learning: Proceedings of the INNS Big Data and Deep Learning Conference INNSBDDL2019, held at Sestri Levante, Genova, Italy 16-18 April 2019* (pp. 78-88). Springer International Publishing.

[6] Bin Sulaiman, R., Schetinin, V. and Sant, P., 2022. Review of machine learning approach on credit card fraud detection. *Human-Centric Intelligent Systems* , *2*(1-2), pp.55-68.

[7] Alghofaili, Y., Albattah, A. and Rassam, M.A., 2020. A financial fraud detection model based on LSTM deep learning technique. *Journal of Applied Security Research*, *15*(4), pp.498-516.Alghofaili, Y., Albattah, A. and Rassam, M.A., 2020. A financial fraud detection model based on LSTM deep learning technique. *Journal of Applied Security Research*, *15*(4), pp.498-516.

[8] Turaba, M.Y., Hasan, M., Khan, N.I. and Rahman, H.A., 2022, October. Fraud Detection During Financial Transactions Using Machine Learning and Deep Learning Techniques. In *2022 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)* (pp. 1-8). IEEE.

[9] Muaz, A., Jayabalan, M. and Thiruchelvam, V., 2020. A comparison of data sampling techniques for credit card fraud detection. *International Journal of Advanced Computer Science and Applications*, *11*(6).

[10] Saputra, A., 2019. Fraud detection using machine learning in e-commerce. *International Journal of Advanced Computer Science and Applications*, *10*(9).

[11] Rubaidi, Z.S., Ammar, B.B. and Aouicha, M.B., 2022. Fraud detection using large-scale imbalance dataset. *International Journal on Artificial Intelligence Tools*, *31*(08), p.2250037.

[12] Sowah, R.A., Kuuboore, M., Ofoli, A., Kwofie, S., Asiedu, L., Koumadi, K.M. and Apeadu, K.O., 2019. Decision support system (DSS) for fraud detection in health insurance claims using genetic support vector machines (GSVMs). *Journal of Engineering*, 2019.

[13] Gangwar, A.K. and Ravi, V., 2019. Wip: Generative adversarial network for oversampling data in credit card fraud detection. In *Information Systems Security: 15th International Conference, ICISS 2019, Hyderabad, India, December 16–20, 2019, Proceedings 15* (pp. 123-134). Springer International Publishing.

[14] Botchey, F.E., Qin, Z. and Hughes-Lartey, K., 2020. Mobile money fraud prediction—a cross-case analysis on the efficiency of support vector machines, gradient boosted decision trees, and naïve bayes algorithms. *Information*, *11*(8), p.383.

[15] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, pp.321-357.

[16] He, H., Bai, Y., Garcia, E.A. and Li, S., 2008, June. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). Ieee.

[17] Han, H., Wang, W.Y. and Mao, B.H., 2005, August. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878-887). Berlin, Heidelberg: Springer Berlin Heidelberg.

[18] Elkan, C.,2001, August. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence* (Vol. 17, No. 1, pp. 973-978). Lawrence Erlbaum Associates Ltd.

[19] He, H. and Garcia, E.A., 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), pp.1263-1284.

[20] Le, T., Vo, M.T., Vo, B., Lee, M.Y. and Baik, S.W., 2019. A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction. *Complexity*, *2019*.

[21] Chen, L., Jiang, J. and Zhang, Y., 2021. HSDP: a hybrid sampling method for imbalanced big data based on data partition. *Complexity*, *2021*, pp.1-9.

[22] Kovács, G., 2019. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, *83*, p.105662.

[23] Batista, G.E., Prati, R.C. and Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), pp.20-29.

[24] "Vehicle Insurance Claim Fraud Detection| Kaggle." https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection Accessed 10 Jan 2023

[25] Breiman,L.,2001. Random forests. *Machine learning*, *45*, pp.5-32.

[26] Ramraj, S., Uzir, N., Sunil, R. and Banerjee, S., 2016. Experimenting XGBoost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, *9*(40), pp.651-662.

[27] Freund, Y. and Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* ,*55*(1), pp.119-139.

[28] Cover, T. and Hart, P., 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), pp.21-27.

[29] Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, *20*, pp.273-297.

[30] Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X., 2013. *Applied Logistic Regression: David W. Hosmer, Stanley Lemeshow, Rodney X. Sturdivant*. Wiley.

[31] Saad Rubaidi, Z., Ben Ammar, B. and Ben Aouicha, M.,2022, December. Comparative Data Oversampling Techniques with Deep Learning Algorithms for Credit Card Fraud Detection. In International Conference on intelligent Systems Design and Applications (pp. 286-296). Cham: Springer Nature Switzerland.

[32] Anguluri Rajasekhar, Ravi Kumar Jatoth, Ajith Abraham, Design of intelligent PID/PID speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm, Engineering Applications of Artificial Intelligence, 29: 13-32, 2014.

[33] Hesam Izakian, Behrouz Ladani, Kamran Zamanifar and Ajith Abraham, A Particle Swarm Optimization Approach for Grid Job Scheduling, Third International Conference on Information Systems, Technology and Management, Communications in Computer and Information Science, Springer Verlag, ISBN 978-3-642-00404-9, pp. 100-109, 2009.

[34] Musrrat Ali, Millie Pant and Ajith Abraham, Simplex differential Evolution, Acta Polytechnica Hungarica, 6(5):95-115, 2009.

[35] Amit K.Shukla, Manvendra Janmaijaya, Ajith Abraham, Pranab K. Muhuri, Engineering applications of artificial intelligence: A bibliometric analysis of 30 years (1988-2018), Engineering Applications of Artificial Intelligence, 85: 517-532, 2019.

[36] Ivan Zelinka, Vaclav Snasel and Ajith Abraham, Handbook of Optimization: From Classical to Modern Approach, Intelligent Systems Reference Series, ISBN 978-3-642-30503-0, Springer Verlag Germany, 1100 p, 2012.

[37] Zahra Pooranian, Mohammad Shojafar, Jemal Abawajy, Ajith Abraham, An efficient meta-heuristic algorithm for grid computing, Journal of Combinatorial Optimization, 30(3): 413-434, 2015.

[38] Prithwish Chakraborty, Swagatam Das, Gourab Ghosh Roy and Ajith Abraham, On Convergence of the Multi-objective Particle Swarm Optimizers, Information Sciences,181(8):1411-1425, 2011.