Digital payment fraud detection methods in digital ages and Industry 4.0

Article *in* Computers & Electrical Engineering · May 2022

DOI: 10.1016/j.compeleceng.2022.107734

5 authors, including:

Victor Chang
Aston University
613 PUBLICATIONS **25,820** CITATIONS
SEE PROFILE

Le Minh Thao Doan
Teesside University
14 PUBLICATIONS **330** CITATIONS
SEE PROFILE

Alessandro Di Stefano
Teesside University
53 PUBLICATIONS **585** CITATIONS
SEE PROFILE

Zhili Sun
University of Surrey
343 PUBLICATIONS **5,936** CITATIONS
SEE PROFILE

Digital Payment Fraud Detection Methods in digital ages and Industry 4.0

Victor Chang[1]*, Le Minh Thao Doan[2] Alessandro Di Stefano[2] Zhili Sun[3]and Giancarlo Fortino[4]

1. Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK
2. Cybersecurity, Information Systems and AI Research Group, School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK
3. Institute for Communication Systems (ICS), 5G&6G Innovation Centre, University of Surrey, Guildford, Surrey, UK
4. Department of Informatics, Modeling, Electronics and Systems (DIMES) of the University of Calabria (Unical), Rende (CS), Italy.

Emails: victorchang.research@gmail.com/V.Chang@tees.ac.uk; minhthaodoanle@gmail.com/L.Doan@tees.ac.uk ; A.DiStefano@tees.ac.uk and z.sun@surrey.ac.uk; giancarlo.fortino@unical.it

*: corresponding author

ABSTRACT

The advent of the digital economy and Industry 4.0 enables financial organizations to adapt their processes and mitigate the risks and losses associated with the fraud. Machine learning algorithms facilitate effective predictive models for fraud detection for Industry 4.0. This study aims to identify an efficient and stable model for fraud detection platforms to be adapted for Industry 4.0. By leveraging a real credit card transaction dataset, this study proposes and compares five different learning models: logistic regression, decision tree, k-nearest neighbors, random forest, and autoencoder. Results show that random forest and logistic regression outperform the other algorithms. Besides, the undersampling method and feature reduction using principal component analysis could enhance the results of the proposed models. The outcomes of the studies positively ascertain the effectiveness of using features selection and sampling methods for tackling business problems in the new age of digital economy and industrial 4.0 to detect fraudulent activities.

Keywords: digital payment; fraud detection; machine learning; Industry 4.0; cybersecurity for Industry 4.0

I. INTRODUCTION

Industry 4.0 has facilitated the rise of e-commerce, leading to the proliferation of digital payment [1]. Physical and IoT devices are connected with digital systems in Industry 4.0, allowing for better collaboration across the ecosystem, enhancing processes, and driving growth [2]. More and more enterprises and industries take into account this new industrial revolution. Unfortunately, keeping pace with this growth, the cybercrime rate in

digital payments is not far behind. Indeed, significant monetary losses every year due to the growing trend of fraudulent transactions urge the financial industry to improve fraud detection systems continuously. There are several control steps in a typical fraud detection process, each of which can either be managed by humans or automated. Unfortunately, the detection is notoriously tricky and influenced by factors such as money and customer spending behaviors. Machine learning algorithms, a leading technological advance for Industry 4.0, have presented promising solutions to tackle these problems by enabling providers to optimize their customer database with real-time transaction details, automatically detecting fraud and verification methods [1, 2].

Over the past decade, there have been extensive machine learning studies to detect fraudulent transactions— one of the popular methods to classify payment transactions is to distinguish fraud using regular labels in database training, also known as supervised learning. The popular methods in this field include k-nearest neighbors (KNN) [3], decision trees [4], logistic regression [5], and support vector machine (SVM) [6]. These learning methods can either be applied separately or assemble models such as random forest to detect fraudulent transactions [7]. The supervised approach utilizes labeled historical transactions to build a predictive fraud model, which returns the likelihood of any new activity being a fraud. Nevertheless, labeling is time-consuming, and not all labels are ready quickly for the learning design [8]. Therefore, there is a light of studies that applied anomaly identification, known as unsupervised learning, to detect fraudulent transactions. This approach aims to discriminate the data pattern of transactions and interpret outliers as fraud. Hence, unseen fraudulent activities can be recognized and do not depend on past labeled transactions. A study in [9] found that unsupervised learnings could handle the skewness issues in this field and give high classification results. Autoencoder, an artificial neural network, appears to be a compelling method in Industry 4.0's fraud detection in recent unsupervised deep learning due to its excellent capability to analyze complex, real-time, and large-scale data [10].

Researchers devote efforts to compare different learning models but mainly focus on the same categorical learnings, such as different supervised learning method comparisons. However, little research has evaluated and compared supervised and unsupervised learning models in the field [11]. This study addresses this gap by evaluating the ability to detect fraudulent patterns of different models ranging from supervised to unsupervised learnings. We propose four supervised algorithms, namely logistic regression, KNN, decision tree, random forest, and an unsupervised algorithm – autoencoder.

Furthermore, identifying the best-fit algorithms is still a critical challenge for researchers in this field for several reasons. First, digital transactions are highly imbalanced with small proportional fraudulent transactions, resulting in inaccurate performance evaluation. Next, finding suitable features, a crucial task to reduce redundant

and irrelevant features from the actual dataset, reducing training time, avoiding overfitting, and improving learning performance are also concerns [12]. Thus, the effects of different sampling methods to solve skewed data and feature selection on the performance of the four proposed supervised learning models are also investigated in this study. The dataset about European credit card holders' transactions [13] is applied to train and test our proposed model in terms of the Area Under the Receiver Operating Curves (AUROC) and average precision.

This study proposes and evaluates machine learning models to detect fraudulent transactions, which is a significant technical matter for financial providers in the Industry 4.0 era. The comparison comprises classification approaches and anomaly detection approach automatically recognizing and distinguishing unusual data from financial databases. By doing so, our contributions are fourfold. First, we contribute to this critical issue and present practical learning algorithms that generate predictive fraud models adapted for financial business in Industry 4.0. Second, our work represents the investigation of feature selection influence, a challenge and critical problem affecting detecting fraud. Third, the imbalanced classification issue, a topic that is usually neglected in previous comparative studies, is resolved in this study by applying more appropriate performance metrics and accessing different effects of oversampling and undersampling techniques. Hence, we can identify an efficient and stable model for fraud detection platforms in the age of Industry 4.0. Finally, this study applies a real transactions dataset to analyze the performance of several learning algorithms to recognize fraudulent patterns. Although Industry 4.0 accelerates information transparency, there is a scarcity of real-world financial records to support the development of a fraud detection systemdue to confidentiality.

Our findings are vital for gaining insights into the robustness of multiple learning techniques in detecting credit card fraud in real-life circumstances. Likewise, as Industry 4.0 is still evolving and new forms of cybercrime and fraud keep emerging, our findings might help the association assemble a vastly enhanced fraud detection system for financial institutions and online payment providers that can better deal with the skewed data and employ more reliable measurements to evaluate the results.

This paper is organized as follows. It starts with an introduction to the topic. The next section reviews relevant studies in Industry 4.0, followed by the methodology section describing the applied methods and workflow of our study. Section 4 presents the experiment and results of the fraud detection model. Finally, the findings are discussed, and the implications are provided in Section 5.

II. INDUSTRY 4.0

Industry 4.0, also referred to as the fourth industrial revolution, was first introduced in Germany in 2011 and officially published in 2013 as a German strategic plan to improve production systems to boost national

business productivity and efficiency. Industry 4.0 draws a new industrial strategy by integrating the organizations to comprehensive technologies related to connectivity, digitalization, and automation [14]. The associated technologies comprise but are not limited to big data, data analytics, artificial intelligence (AI), the internet of things (IoT), and cyber-physical systems (CPS). Industry 4.0's goals are to promote operational effectiveness and efficiency and achieve a greater level of automatization.

Industry 4.0 has an undeniable influence on almost every aspect of our daily lives, affecting and changing the ways industries work and how individuals and organizations interact with technology. In particular, the digital transformation has extensively impacted financial industries in the age of Industry 4.0, with a massive amount of digital transactions undertaken every day. Technology advancement for Industry 4.0 has transformed traditional banking into financial technology.

Business works steadily rely on data, and with increasing hacking and unauthorized access, information systems have become more vulnerable. Hence, cybersecurity has become an escalating challenge regarding the top business risks and plays a leading role in maintaining business competitiveness within Industry 4.0 [15]. Fraud detection is a prominent part of cybersecurity in the Industry 4.0 era [2]. Therefore, strengthening fraud detection systems and cybersecurity efforts is essential for any financial institution or payment provider in the new digital age. Data science and AI play a significant role in improving cybersecurity, particularly in the digital payment fraud detection field. As a result, integrating data science and AI with industry 4.0 will assist financial institutions in increasing the effectiveness and efficiency of automated processes detecting and preventing fraudulent activities.

III. METHODS

Exploratory Data Analysis (EDA) was applied to explore and understand data. The flowchart of the data process is illustrated in Figure 1. First, the data was converted into a data frame during the preprocessing stage and checked for miss and duplicated values were. Then, data was prepared for training by normalizing features, deleting unused columns, and getting a sample. Because this data was highly imbalanced, 5,000 transactions were randomly re-sampled to make the training more efficient and save time. After that, sample data were split into training and test sets with a ratio of 80:20. Models were designed, then models were evaluated by training and test sets. A set of experiments have been conducted and compared using performance metrics such as AUROC and precision. Other performance metrics such as accuracy, sensitivity, and specificity were also provided to support the comparison.

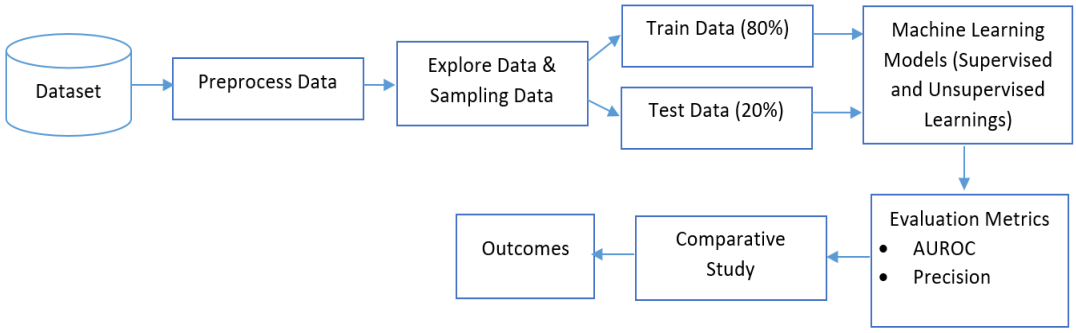


Figure 1: Flowchart of the data process

This section explains the dataset adopted in the study and the five proposed algorithms, specifically logistic regression, KNN, decision tree, random forest, and autoencoder. The following methods are also applied to the proposed model for supervised learning algorithms. First, the GridSearchCV method is applied with 5-fold cross-validation to tune and find the optimal hyperparameters and reduce overfitting. Then, Principal Component Analysis (PCA) extracts features, while the Synthetic Minority Oversampling Technique (SMOTE) and Near Miss Undersampling methods reconstruct imbalanced training data into a balanced one. The autoencoder model is chosen for the unsupervised learning algorithm because of its ability to handle imbalanced data very well. It can be trained using one class of data without the data labels. Different epochs and batch sizes are tried to find the optimal solution.

1. Dataset Description

This study uses the dataset of September 2013 credit card transactions provided on Kaggle [13]. This dataset is widely used to test the proposed learning model by many studies [9], [16] to detect fraudulent transactions. It comprised 284,807 transactions, of which 492 were fraudulent. Because of the confidentiality, the essential information and detailed features names about the dataset were not revealed. PCA was used to transform the other 28 features except for the "Time" and "Amount" features. The "Time" feature was given by the seconds elapsed between the first transaction and the current transaction, while "Amount" is the money of the cardholder's purchase. Data was binary classified, and one corresponds to a fraudulent transaction, while zero is a regular one.

2. Logistic Regression (LR)

Logistic regression is used for the classification problem by gauging the outcome probability of a particular class. The predictions are transformed using the logistic function, which returns probability values between 0 and 1. Logistic regression is widely applied for classification tasks [17] and widely adapted for Industry 4.0 because it is easy to implement, has good accuracy, and is very efficient to train. Itoo et al. [5] indicated that logistic regression achieved the best performance in fraud detection compared to Naïve Bayes and KNN.

Different combinations of algorithms such as "liblinear," "sag," and "lbfgs" and regularisation strength effects (C values) were employed to find out the optimal method.

- Lbfgs stands for Limited-memory Broyden–Fletcher–Goldfarb–Shanno and is the defaulted solver in the Sklearn Logistic Regression model. This solver used the inverse Hessian matrix, or second partial derivatives calculations, to update the gradient evaluations and discard earlier gradients, making the computation more effective and memory-saving.
- Liblinear is a linear classification algorithm supporting logistic regression and is claimed to work well with small datasets. It used a coordinate descent algorithm and successively performed approximate minimization along coordinate to optimize the problems.
- Sag is Stochastic Average Gradient Descent and a breakthrough method in stochastic optimization that significantly reduced variance. It is an iterative technique to optimize the sum of a finite number of smooth convex functions. Although this method is required to maintain the table of gradients and the average gradient values, it is very efficient in terms of time and is usually applied for large datasets.
- C measures the inverse power of the regularisation influence, i.e., the smaller the value of C is, the greater the regularisation effect. There is a trade-off between low-bias and high-variance, so identifying the appropriate C value is critical to optimize the model. The C values of 0.01, 0.1, 1, and 100 were used in the model design.
- L2 is named Ridge Regression and used as a regularizer. It adds the "squared magnitude" of coefficients depending on the model complexity, so an additional component will be added to penalize the loss function.

The Ridge Regression (l2) can be obtained by maximizing the likelihood function with a penalized parameter applied to all the coefficients except the intercept. For a given i instance in an n observations dataset, $x_{i,}, y_i$ are the feature input and output. The parameters to maximizing the log-likelihood function can be expressed as below

$$l(\beta) = \sum_{i}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i) = \sum_{i}^{n} [y_i x_i \beta - \log(1 + e^{x_i \beta})]$$

where, β is the regression coefficients. The coefficients estimates are the values that maximize the following slightly different log-likelihood function where a ridge penalty (l2) is added to the function. Then, the objective function of the logistic regression algorithm can be written as

$$l(\beta) = \sum_{i}^{n} [y_i x_i \beta - \log(1 + e^{x_i \beta})] - \lambda \sum_{j=1}^{p} \beta_j^2$$

where, λ is the hyperparameter and p is the weight of β. In other words, l2 regularization is a quadratic function of the weight values. By adding regularization, the model can be optimal and reduce overfitting.

3. k-Nearest Neighbour Classifier (KNN)

KNN is non-parametric and instance-based learning based on similarity measures and grouped based on a similar manner. First, the distance between the current and input data points is calculated for every data point and is sorted in ascending order. Then, the prediction is composed by averaging the result, and k items with the smallest distances are chosen. Finally, the most common class among the k nearest neighbors is assigned, and a majority vote of its neighbors classifies the input data point. KNN is a classifier with only a proximate local function, and all its calculation is delayed until evaluating the model. The KNN algorithm is selected because it does not need any training for model generation, so its accuracy does not impact even new data added, suitable for real-time fraudulent detection. A comparative study found that KNN performs better than other algorithms such as Naïve Bayes to detect fraud on real-life payment transactions [3]. However, it might be costly in terms of time and memory when working on an extensive dataset.

In this study, different metrics such as "Minkowski," "Euclidean," and "Manhattan," and parameter tuning for k from 2 to 7 are tried to find the optimal model.

- "Euclidean" Distance is the shortest distance between two points and is measured as the square root of the total of the squared differences between two points. The distance can be expressed as $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
- "Manhattan" Distance is measured as the total of the absolute differences between two points, which is calculated as $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$
- "Minkowski" Distance, a generalized form of the two above distance metrics, is computed as $d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$, where p represents the order of the parameter.

4. Decision Tree (DT)

A decision tree, one of the most widely used algorithms, is a non-parametric supervised learning, where data is continuously divided based on a specific rule [18]. It builds in the form of a tree structure with a set of if-then-else decision rules. It begins with a root node, a decision tree, splits into separate branches, connects with other nodes, and so on. The outcome is a tree with leaf nodes and decision nodes. Therefore, the decision tree can isolate a complicated issue into a simple one due to this tactical strategy of splitting and deciding. The significant

advantage of a decision tree is its interpretability and quickly draw insights from the modeling process flow by depicting output visualization. Besides, the decision tree is fast and does not need much effort in data preprocessing like scaling data. Sahin and Duman [18] indicated that the decision tree model outperforms Support Vector Machine (SVM) to solve fraud detection problems. However, the decision tree is easy to overfit if a small sample is used.

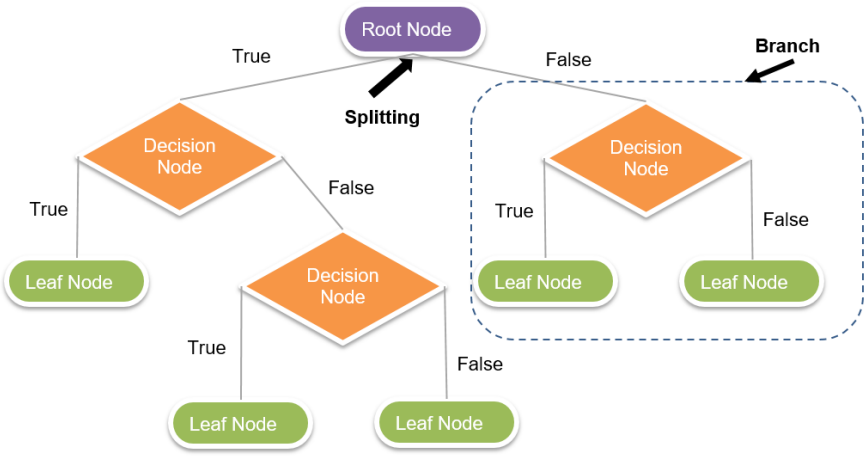


Figure 2: Decision Tree diagram

Different combinations of criterion functions such as "Gini" and "Entropy," the minimum number of splitting samples from 2 to 7, and the maximum depth of the tree from 2 to 6 were used to find the best-fit model. The Gini impurity ("Gini") and entropy are the functions to measure the quality of a split in the decision tree algorithm. Precisely, the Gini impurity measures the frequency at which a specific node wrongly classifies the randomly chosen element and is defined as

$$Gini = 1 - \sum_{i=1}^{N} (p_i)^2$$

where N represents the total number of output classes in the set, and p_i represents the conditional probability that the target variable is in class i in the set.

Entropy indicates the randomness or the disorder of the features with the target and plays an essential role in calculating Information Gain (IG). For each attribute, the entropy is computed using the below formula

$$Entropy = 1 - \sum_{i=1}^{N} p_i \times \log_2 p_i$$

Then IG can be measured based on entropy as $IG = entropy\,(parent) - entropy\,(children)$. Like Gini impurity, this measure is used to decide which feature to split on at each step in building the tree. IG favors smaller

partitions, while Gini usually favors larger partitions. Therefore, Gini impurity is calculated with less computation, and it is easy to implement in practice.

5. Random Forest (RF)

As stated above, the decision tree is easy to overfit and sensitive to particular data [19]. Ensemble methods can unravel these issues by combining predictions from multiple trees to make more accurate predictions than a single one. Today, the random forest model is one of the most powerful ensemble models based on decision trees and the bagging mechanism. Bagging refers to trained multiple decision trees on various subsamples of data called bootstrapping and then followed by aggregation. It grows various classification trees, such that for each built tree, random data are run down the tree to compute the proximity for each pair of cases. As such, each tree in the forest is unique and has the same distribution. The random forest algorithm usually shows a great generalization as it aggregates different trees' decisions. Moreover, it also avoids the overfitting problem of complex decision trees. Furthermore, as each tree is built independently, random forest is computationally efficient and robust to outliers. Its application has been popular in recent years, especially in fraud detection, because it is easy to use and has high-performance results [20].

Different combinations of criterion functions, such as "Gini" and "Entropy," the minimum number of splitting samples from 2 to 7, and the maximum depth of the tree from 2 to 6 were used to find the best fit the random forest model.

6. Autoencoder (AE)

AE is deemed a compelling model in unsupervised learning for Industry 4.0. It is an exceptional neural network architecture whose output is the same size as the input. An AE has two phases: an encoder generating a compressed coding of the training data and a decoder reconstructing the given input, as illustrated in Figure 3. Normal data generally train AE for anomaly detection to obtain the reconstruction error (RE). RE is the difference value between the reconstructed and the initial variant. Normal data's RE is assumed to be more negligible as it is similar to the learning data, while the abnormal data one should be higher. Mean squared error (MSE) is applied in this study as RE and is computed as the following formula.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

Where n is the number of data points, and Y_i and $\hat{Y}_i$ are observed and predicted values, respectively.

Autoencoder was performed well on imbalanced data and outperformed LR in fraud detection [21]. Besides, unlike supervised learning, it demonstrated the ability to detect fraudulent transactions without effort in

feature engineering, hence, saving time [22]. Therefore, this study proposed an Autoencoder model to evaluate its ability to detect fraud by different batch sizes and learn rates.

Both encoder and decoder layers apply ReLU activation functions, lately rising activation function to reduce training loss and improve training time. In addition, L2, or Ridge regression, is applied as a regulariser to avoid overfitting issues. Furthermore, Adam optimizer, an extension of Stochastic Gradient Descent, is selected in the model compile.

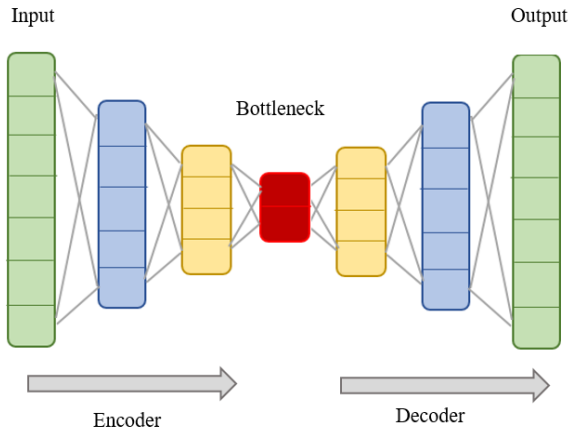


Figure 3: Autoencoder Model

7. Sampling

The synthetic minority over-sampling technique (SMOTE) is extensively discussed among the oversampling method [23] and is widely applied to deal with skewness in fraud detection issues [20]. Oversampling adjusts the data to increase the number of minorities to balance with the majority. It over-samples the minorities and creates synthetic samples using k-NN. Once the minority class sample x is selected amount k neighbor, the synthetic sample x_{new} generated by interpolating between x and $\tilde{x}$ is illustrated in Appendix 1 and follow the formula (3):

$$x_{new} = x + rand(0,1) \times (\tilde{x} - x) \quad (3)$$

Where rand(0,1) is a random number between 0 and 1

SMOTE improves the model's reliability, convergence speed, accuracy, and efficiency [17]. However, this approach had some implications, such as the potential overfitting issues, and the generated data might not resemble the initial data.

Undersampling appears to mitigate the above oversampling issues. For this technique, the majority class instances are randomly selected and added to the minority class with a 1:1 ratio. NearMiss is a popular undersampling technique to handle imbalanced data [24]. The example of how NearMiss works is presented in

Appendix 2. However, NearMiss could discard potentially helpful information essential for building rule classifiers and introducing bias to the training set like other undersampling methods. Therefore, in many cases, oversampling tends to yield better accuracy and is widely used in fraud detection.

8. Feature extraction

Feature extraction is performed to save computational resources and training time. Besides, it has been found to avoid overfitting and potentially boost the algorithms' performance. Feature extraction is the approach where raw data is transformed into features while preserving the information in the original dataset. Principle Component Analysis (PCA) is a widely applied feature extraction method in research for reducing the dimensionality of the data, optimizing the machine resources while minimizing information loss. The data dimensionality is reduced by creating new features from the existing dataset. PCA has been applied in many research domains, including machine learning and pattern recognition, for extracting optimal features [25]. Therefore, it is applied in our study to investigate the feature extract effect on models' performance.

According to Song et al. [25], PCA involves four following steps:

- PCA's correlation/covariance matrix and the eigenvalues and eigenvectors are computed;
- The eigenvectors $V_1, \ldots, V_n$ corresponding to the first n largest eigenvalues are selected ;
- The feature extraction result's contribution of the k^{th} feature component is measured as follows:

$$c_k = \sum_{i=1}^{n} |V_{ki}|$$

- The principal components c_k is selected and sorted in descending order.

By doing so, a new dataset, constructed by transposing the eigenvectors and the initial data, is the cross-product of these two matrices. It decreases the number of features by generating a new and smaller number of features that capture a substantial amount of the information in the original data.

IV. DATA STRATEGY AND EXPERIMENTS

The study's data strategy included five stages: loading and preprocessing data, exploring data, preparing data for learning model, training data, and evaluating results. Python 3.9 and its supported libraries were used for data strategy, particularly NumPy and Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and Sklearn and Tensorflow for machine learning. Five experiments were evaluated using AUROC and average precision score to find the best performance model.

1. Data loading and preprocessing

The credit card transaction dataset was downloaded on Kaggle and converted to a data frame for further processing. There was no missing value observed in this data; however, 1081 duplicated values were found and removed.

2. Exploratory Data Analysis (EDA)

Then, EDA was carried out to explore the data. There were more than 283,000 transactions in this dataset, and it was highly imbalanced with only 0.17% (473 transactions) fraud; hence, re-sample the data was recommended to train efficiently. Data comprised 30 features, including Time, Amount, V1, to V28, which distribution was visualized in Figure 4. Compared to the regular transactions, fraudulent transactions usually had different distributions across most of the features in this dataset. Besides, Figure 5 illustrates that the transactions' amount in the case showing outliers and without outliers. It was observed that the amount feature was right-skewed, and regular transactions typically have lower amounts than fraudulent ones.

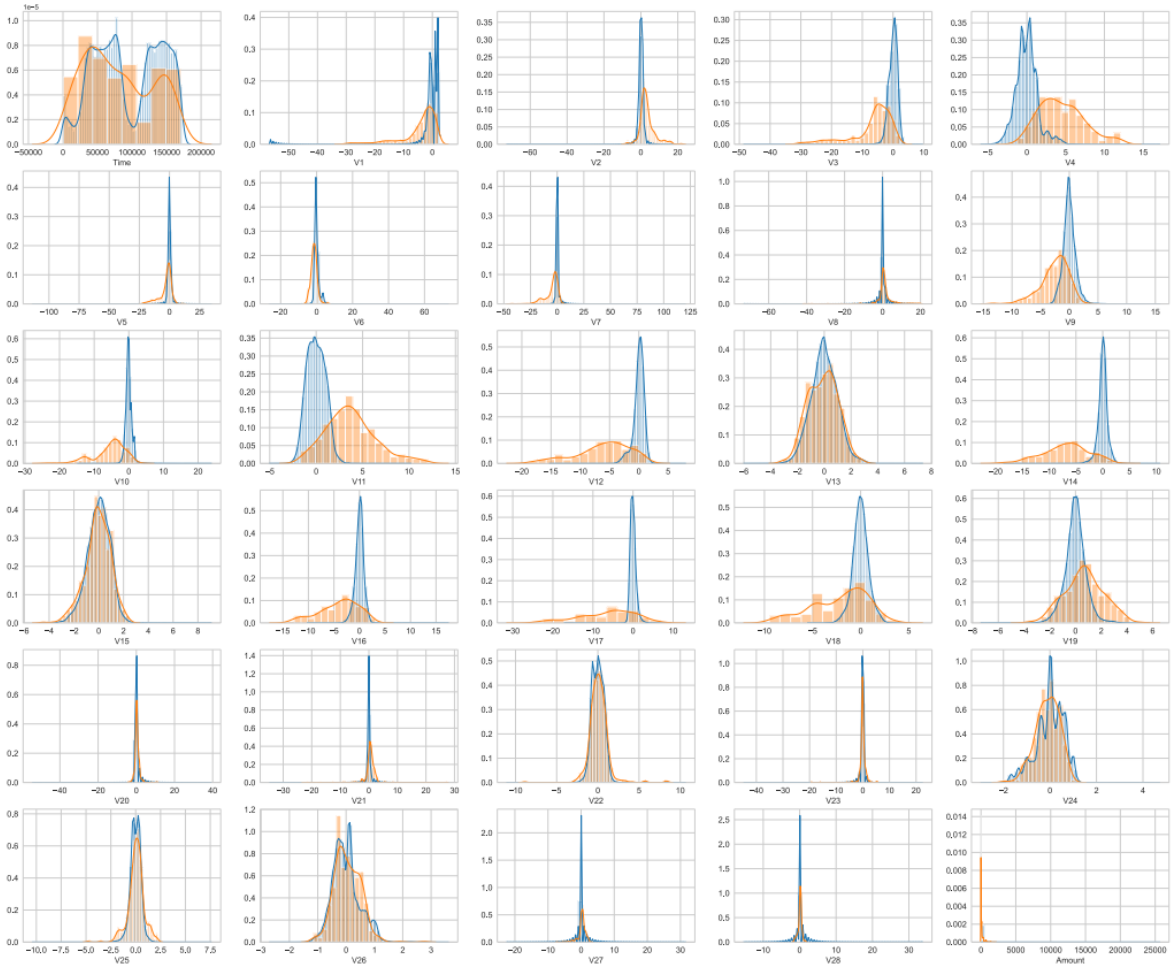


Figure 4: Features distribution with feature name on the x-axis and density on the y-axis. The red color presented fraudulent transactions, and the blue color presented regular transactions.

Furthermore, the feature correlation analysis was performed to see the relationship between variables, as displayed in Figure 6. The intensity of the color shows the associated value of the relationship. Dark colors illustrated positive correlations, while brighter colors presented negative correlations. Except for Amount and V7 features, no highly-correlated variables were observed in this data.

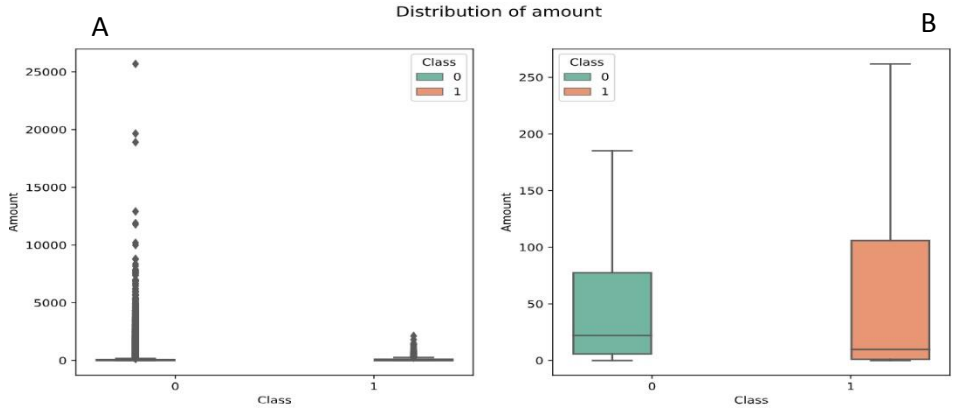


Figure 5: Fraudulent and regular transactions amount distribution including outliers (see A on the left) and excluding outliers (see B on the right).

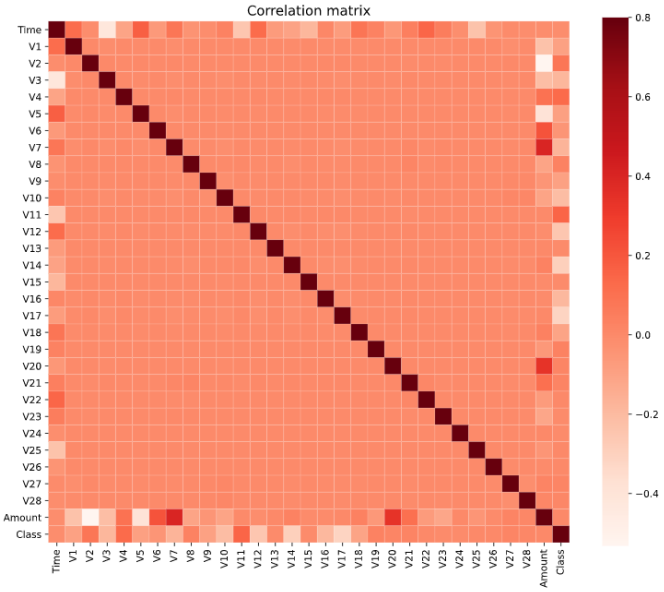


Figure 6: Correlation matrix

3. Data preparation for ML:

A total of 5473 random samples were chosen, including 5,000 random regular transactions and all 473 fraudulent transactions. "RobustScaler" normalized the "Amount" feature to avoid outliers according to the first and third quantile range and get a better model's generalization. This normalization could transform the instance *i* following the below formula:

$$\frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}$$

where Q_1, Q_2, and Q_3 represent the first, second, and third quartiles, respectively.

Then, the "Time" column was removed due to non-relevance. Once done, the new sample data was visualized to check the distribution and feature correlation (Appendix 3,4). Then, sample data were split into train and test sets in a ratio of 80:20. Undersampling and oversampling techniques were employed to convert skewed training data to a balanced one for supervised learning models.

4. Experiments:

Five different experiments were designed to train and evaluate models. Different approaches were applied to tackle the challenges in fraud detection and improve classification performance. Four supervised learning algorithms were used on the sample data in the baseline model (experiment 1), then adjusted the algorithms with class weight to handle imbalanced data (experiment 2). Under-sampled and oversampled were applied to training data in experiments 3 and 4. All supervised models had used 5-fold cross-validation and tuned hyper-parameters by grid search (GridSearchCV) to choose the best estimators. PCA extracted ten first principle components applied to all supervised algorithms to avoid overfitting and improve training efficiency along with the original features. Experiment 5 performed an unsupervised algorithm with different batch sizes and learning rates.

Finally, the experimental results were compared and evaluated using different measures. Although accuracy is the most popular measure for the classification problem, it is not enough to measure the model performance when the data is significantly imbalanced. For example, the models could get high accuracy by focusing on the regular transactions (the majority) and ignoring the fraudulent ones. Therefore, we used two widely applied performance measures for imbalanced data: the Area Under the Receiver Operating Curves (AUROC) and average precision as the primary performance measures to compare the model results.

AUROC indicates the ability to distinguish between classes. In other words, the higher AUROC, the better model will correctly predict the fraud (positive class) and regular (negative class) transactions. In order to compute AUROC, we first plot the receiver operating characteristic (ROC) curve, which is the graph of true positive rate (sensitivity) against false positive rate (1 – specificity). Then, AUROC is closely related to the Gini coefficient and calculated using the trapezoidal rule to estimate the area. It takes values from 0 (inaccurate) to 1 (perfect accurate), where a value of 0.7 – 0.8, 0.8 – 0.9, more than 0.9 suggest acceptable, excellent, and outstanding results, respectively.

$$AUROC = \frac{G_1 + 1}{2} \quad (4)$$

Where $G_1 = 1 - \sum_{k=1}^{n}(X_k - X_{k-1})(Y_k - Y_{k-1})$ (5)

Precision is the ratio of accurate predictive positive class (fraud transaction) to all predicted positives and estimates the accuracy of the fraud predictions of the model. It ranges from 0 (no precision) to 1 (perfect precision) and is measured as the formula below.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

The confusion matrix (Table 1) is applied to estimate the effectiveness of a classifier and consists of four elements, namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN). In our case, TP is the number of fraud accurately predicted, while FP is non-fraud transactions miscategorized as fraud. Similarly, TN is the normal transaction accurately predicted, and FN is the fraudulent transaction misrepresented as normal.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 1: Confusion matrix

In addition, other measures derived from the confusion matrix such as accuracy, sensitivity, specificity are also provided. Accuracy is the ratio of accurate fraudulent prediction executed by the algorithm to the total observations to determine its effectiveness in distinguishing the fraud transaction. Sensitivity and specificity measure the proportion of correctly identified actual fraudulent transactions (positive class) and regular transactions (negative class). Their values range from 0 to 1, where a value of 0 indicates the worst result and 1 indicates the best one.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (8)$$

$$Specificity = \frac{TN}{TN+FP} \quad (9)$$

V. RESULTS

Before starting the experiment, t-SNE and PCA for the first two components were used to visualize the essence of fraudulent and regular transactions. t-SNE, namely t-distributed stochastic neighbor embedding, is a nonlinear dimensionality reduction technique used for data visualization and mapped to a lower-dimensional

space (typically two-dimensional plane) from the multi-dimensional data to identify patterns in the details. This algorithm is widely applied to visualize high-dimensional datasets and is claimed as the best algorithm for 2D visualization in this field. In the t-SNE model, data points are converted based on the joint probabilities, so nearby points are represented by similar samples, while different samples represent distant points. Therefore, it is able to excellently capture the high-dimensional data's local structure and reveal global data structure, such as clusters' existence.

As shown in Figure 7, the data points reflect credit card transactions in a scatter plot, in which regular and fraudulent transactions are represented in blues and red, respectively. The two observed clusters identified by t-SNE are plotting on the right chart, while those identified by PCA are on the left. As shown in the below figure, t-SNE can distinguish between fraudulent and regular transactions, although there are some similar points. In contrast, in the PCA chart, many regular transactions are very similar to fraudulent ones. Therefore, t-SNE, in this case, illustrates better performance in fraud recognition than PCA. Besides, some fraudulent samples are still mixed in regular ones in the chart below, which can be considered a trade-off of reducing information from the dimensional reduction technique. Consequently, we conducted five experiments to investigate the fraud detection ability from our proposed models.

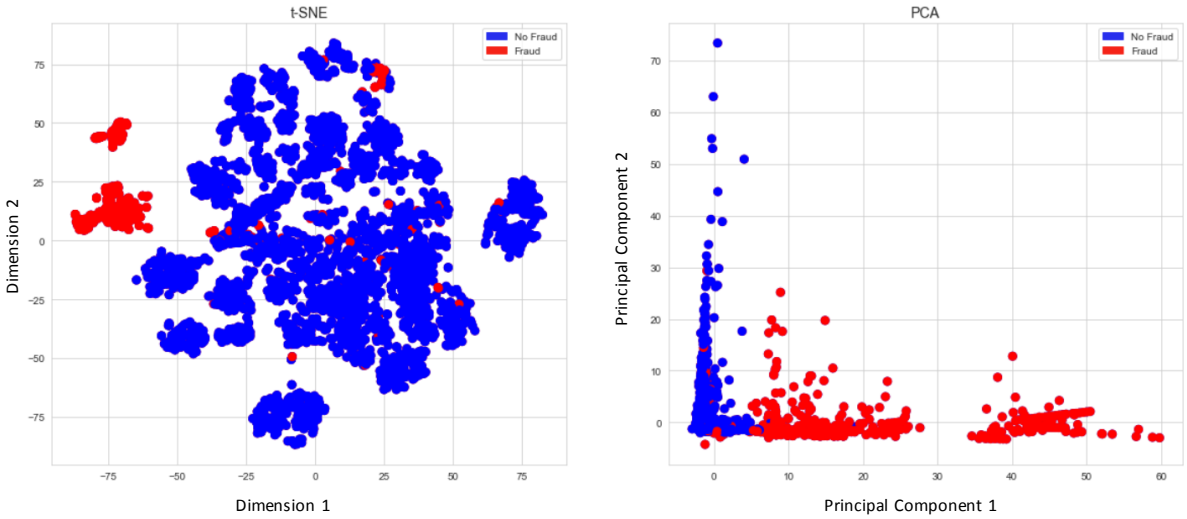


Figure 7: t-SNE and PCA visualization

1. Experiment 1 – Baseline

Figure 8 presents the optimal decision tree visualization for fraud detection using the training set. The depth of this tree analysis is 5, with a minimum 6 number of samples required at a leaf node. The node id number is also provided to distinguish each node. Besides Gini impurity, the probability that the transactions are incorrectly classified according to the subset labels' distribution is applied to evaluate the quality of splits. Gini

value ranges from 0 to 1, where the lower Gini is, the higher chance of the homogeneity of the nodes will be. The decision tree starts from the root node with V14 features not exceeding -3.6 as the splitting rule. Out of 4378 training samples at this node, 314 samples satisfied this rule go to the next decision node 1, while samples that failed this rule go to decision node 14. This node is labeled as fraud with the Gini value of 0.156. Similarly, the process continues to recur on each subset, in which only unselected attributes from the above nodes are considered.

The performance results of four supervised learning algorithms in original features and PCA generated by classification reports are shown in Table 2. The precision-recall (PR) curve and ROC curve showing average precision scores and AUROC values of the four algorithms are also illustrated in Figure 9 to compare the performance of the classifiers. Excellent average precision values are observed from the chart as the closer to the upper right concern PR curve is, the better its value is. Besides, as shown in Figure 9, the ROC curve above the diagonal line indicates that all algorithms reasonably discriminate to predict fraudulent and normal transactions. Besides, the AUROC value is maximized when the ROC reaches the upper left corner, so this experiments' chart reveals a very good model performance. In fact, all AUROC values derived from the chart are greater than 0.9, demonstrating the outstanding test results.

For the original features, RF yields the best results in terms of AUROC and average precision with the value of 0.969 and 0.910, respectively. DT has the worst value of AUROC, while KNN has the worst value of precision. Additionally, PCA has improved the performance of LR and KNN. Consequently, a combination of LR and PCA results in the best performance overall, with an AUROC value of 0.971. DT gets the worst AUROC results in both cases; however, DT is usually applied in practice because of its interpretability by visualizing the operational flow.

For other performance metrics, the four classifiers in experiment 1 all have accuracy values of more than 0.98 and specificity values of 1, while the sensitivity values range 0.82-0.86. Hence, experiment 1 has indicated excellent results; however, these models are more likely to predict the majority favorably when looking into sensitivity and specificity values. The primary purpose is to predict fraud effectively and adapt it for Industry 4.0, so action to tackle the above problem is implemented by adjusting the weight of each class, as in experiment 2.

	Experiment 1	**Accuracy**	**AUROC**	**Average Precision**	**Sensitivity**	**Specificity**
Original features	LR	0.98	0.950	0.909	0.84	1.00
	KNN	0.99	0.905	0.824	0.85	1.00
	DT	0.98	0.902	0.839	0.82	1.00
	RF	0.98	0.969	0.910	0.83	1.00
PCA	LR	0.99	0.971	0.927	0.84	1.00
	KNN	0.99	0.933	0.865	0.86	1.00

	DT	0.98	0.900	0.858	0.83	1.00
	RF	0.98	0.960	0.921	0.83	1.00

Table 2: Experiment 1 Performance Results

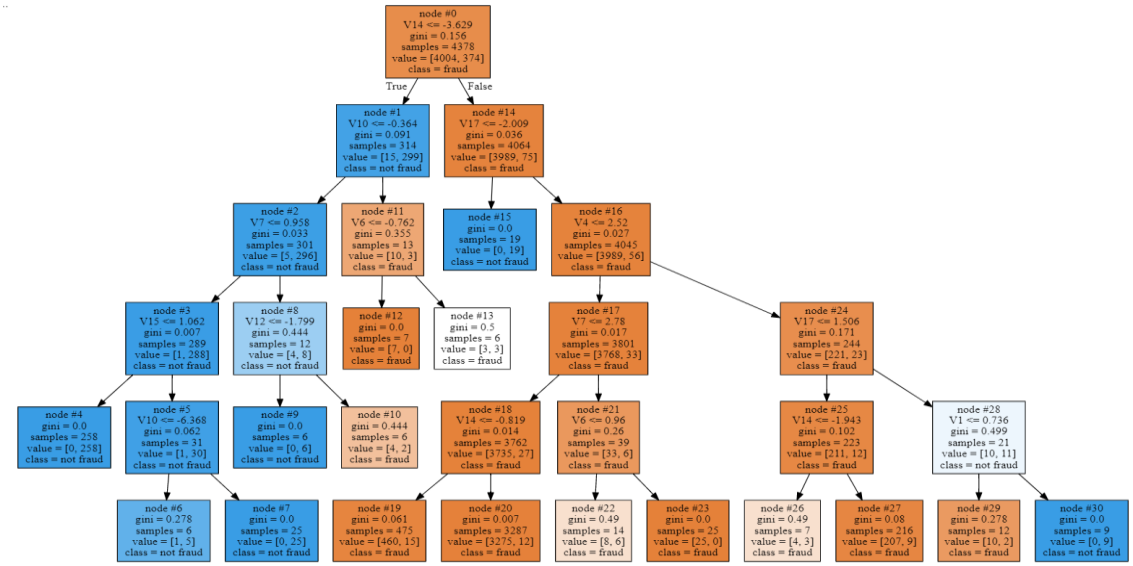


Figure 8: Experiment 1 Decision tree visualization

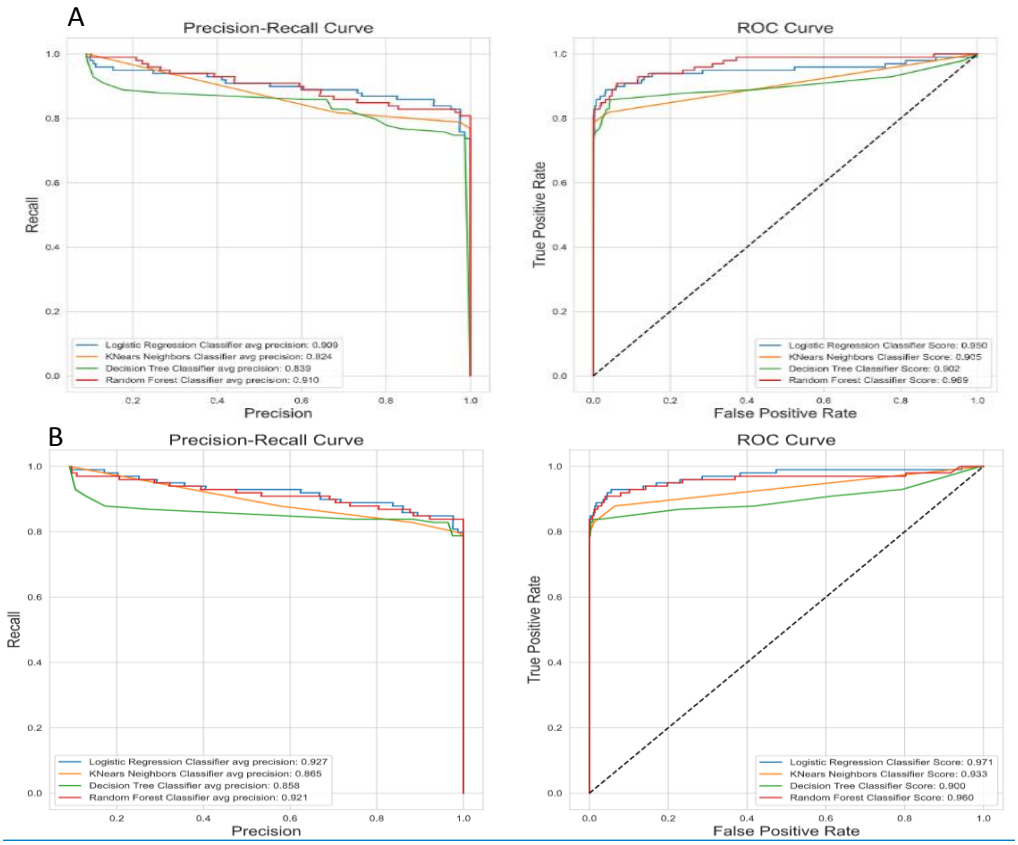


Figure 9: Experiment 1 Precision-Recall Curve and ROC Curve. (A) Original features case, (B) PCA

2. Experiment 2 – Adjusted class weights

The ratio of regular to fraudulent transactions 10.71 was adjusted to the minority class as class weight. As presented in Table 3, RF has the highest AUROC value of 0.957, while LR has the highest average precision value of 0.915 for the original features. Features selection models improve the average precision values and AUROC values of most algorithms. Overall, RF using PCA shows the best result in experiment 2: the AUROC of 0.969 and an average precision score of 0.925. Moreover, it is observed that compared to experiment 1, the adjustment of class weight in experiment 2 could improve the performance of supervised algorithms as well as enhance the sensitivity of the models, which measures the proportion to detect fraud correctly. See Figure10.

	Experiment 2	**Accuracy**	**AUROC**	**Average Precision**	**Sensitivity**	**Specificity**
Original features	LR	0.98	0.955	0.915	0.89	0.98
	KNN	0.99	0.905	0.824	0.85	1.00
	DT	0.97	0.952	0.859	0.87	0.98
	RF	0.98	0.957	0.906	0.84	1.00
PCA	LR	0.98	0.966	0.919	0.88	0.98
	KNN	0.98	0.933	0.865	0.84	1.00
	DT	0.98	0.939	0.870	0.84	0.99
	RF	0.98	0.969	0.925	0.85	1.00

Table 3: Experiment 2 Performance Results

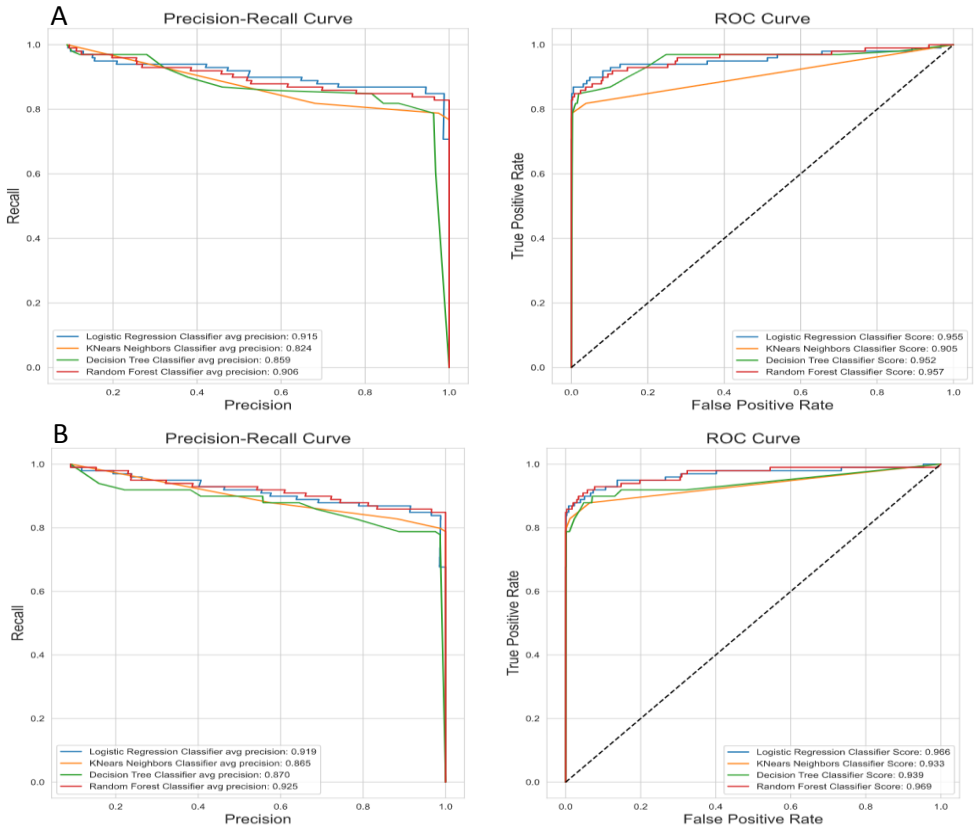


Figure 10: Experiment 2 Precision-Recall Curve and ROC Curve. (A) Original features case, (B) PCA case

3. Experiment 3 – NearMiss Undersampling

NearMiss Undersampling technique is applied in experiment 3. Table 4 highlights that RF applying PCA has the highest AUROC values while LR got the highest precision score in both cases. Among the four models, the performance of RF is the most powerful, while KNN and DT have the lowest results in full features and PCA. Except for the RF model, PCA does not show the performance enhancement in this experiment. See Figure 11.

	Experiment 3	Accuracy	AUROC	Average Precision	Sensitivity	Specificity
Original features	LR	0.98	0.972	0.928	0.82	1.00
	KNN	0.98	0.909	0.836	0.84	0.97
	DT	0.98	0.916	0.842	0.82	0.98
	RF	0.98	0.952	0.896	0.83	1.00
PCA	LR	0.98	0.970	0.927	0.81	1.00
	KNN	0.98	0.909	0.831	0.83	1.00
	DT	0.98	0.903	0.834	0.83	1.00
	RF	0.98	0.976	0.927	0.83	1.00

Table 4: Experiment 3 Performance Results

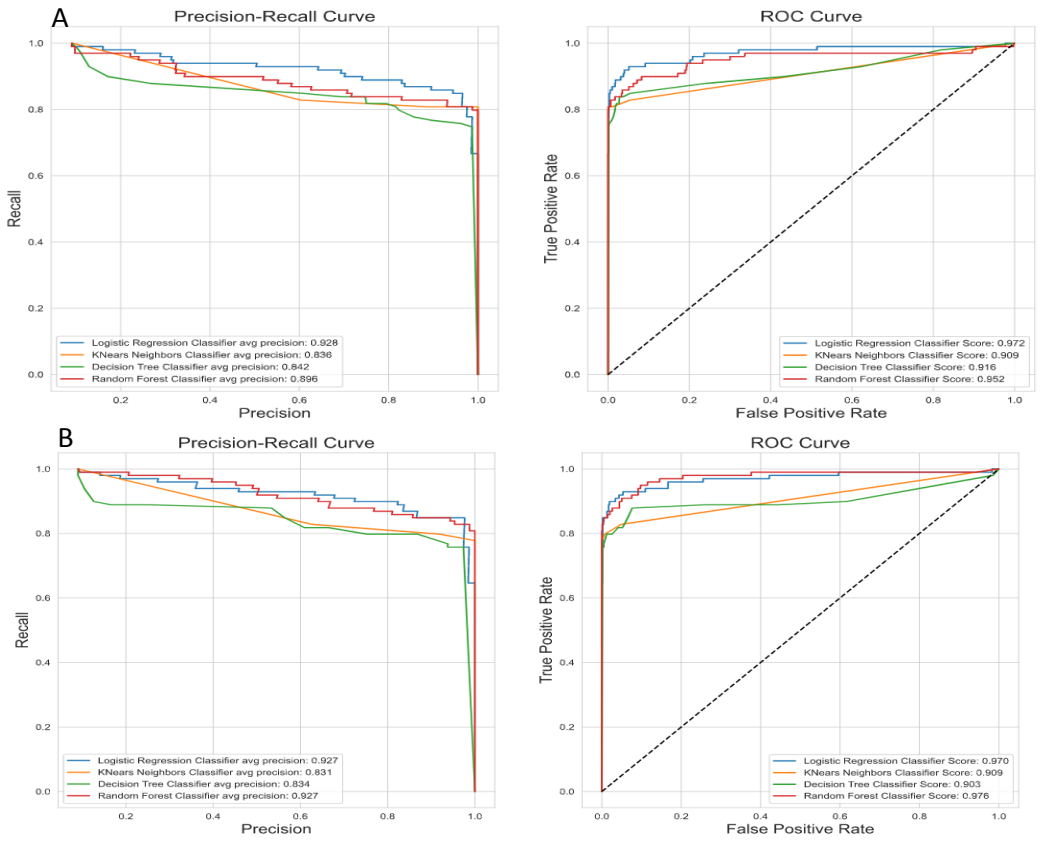


Figure 11: Experiment 3 Precision-Recall Curve and ROC Curve. (A) Full features case, (B) PCA case

4. Experiment 4 – SMOTE Oversampling

SMOTE Oversampling is implemented in experiment 4. Table 5 indicates that RF algorithm results are in the best performance in full features cases and PCA. In addition, AUROC, average precision score, and

sensitivity are significantly enhanced when employing PCA with the oversampling method. Overall, RF using PCA gives the best outcomes: the AUROC of 0.972 and an average precision score of 0.921, while KNN and DT are still the lowest results in experiment 4. See Figure 12.

	Experiment 4	Accuracy	AUROC	Average Precision	Sensitivity	Specificity
Original features	LR	0.97	0.930	0.883	0.90	0.98
	KNN	0.98	0.902	0.825	0.85	0.99
	DT	0.97	0.871	0.718	0.85	0.98
	RF	0.98	0.955	0.898	0.85	1.00
PCA	LR	0.97	0.948	0.906	0.88	0.98
	KNN	0.98	0.905	0.805	0.87	0.99
	DT	0.96	0.931	0.878	0.90	0.97
	RF	0.98	0.972	0.921	0.87	0.99

Table 5: Experiment 4 Performance Results

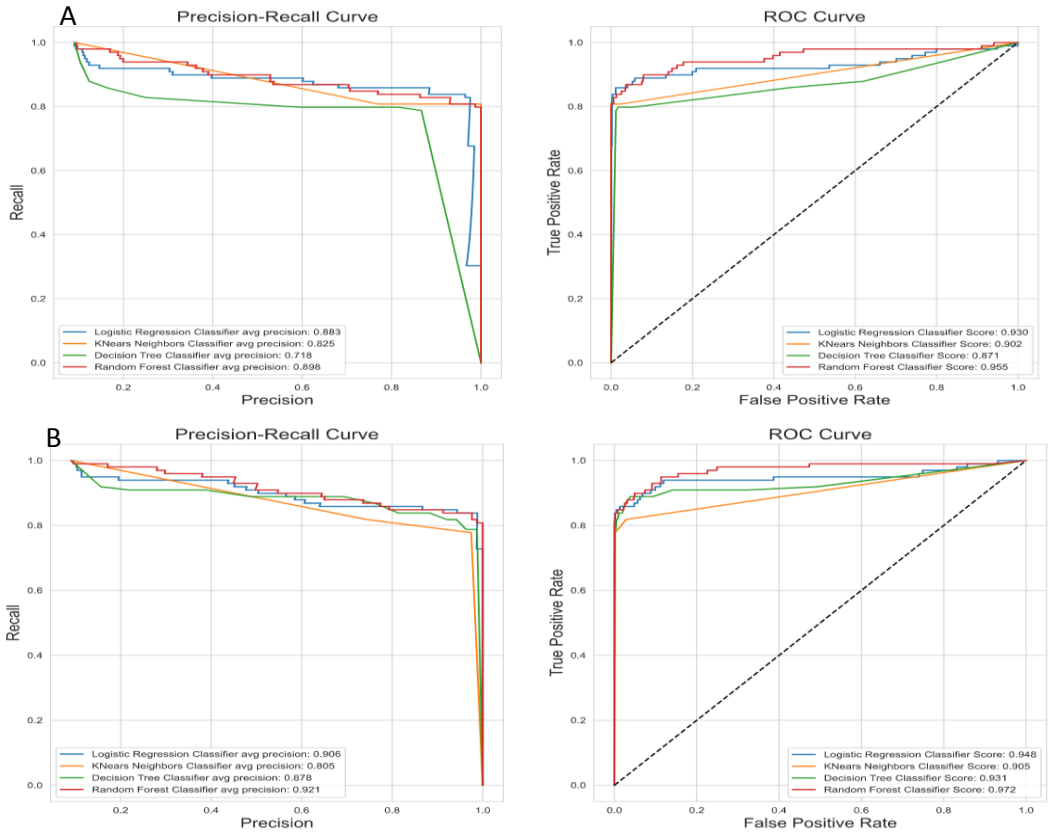


Figure 12: Experiment 4 Precision-Recall Curve and ROC Curve. (A) Full features case, (B) PCA case

5. Experiment 5 - Autoencoder

An autoencoder model is designed with two encoder layers with 14 and 7 nodes, respectively. Different combinations of batch sizes and the learning rate are tried to get the optimal result. Table 6 illustrates that the decrease in batch size and increased learning could improve model performance in the AUROC value.

Furthermore, it is observed that the regular class's MSE value ranges from 0.867 – 0.904, while the MSE value of fraud is much higher, ranging from 4.976 – 5.010. Therefore, as shown in Figure 13, a flexible threshold of 3 is chosen to identify the fraud transaction. We also reconstruct errors using MSE for different classes and plot in Figure 13. The model detects anomalies as points where the reconstruction error is greater than a threshold of 3 because the normal transactions are likely to have lower error values. Therefore, data points with higher errors tend to indicate they are fraud data with higher possibilities, and thus saving more effort in finding fraudulent data. For overall results, experiment 5b indicates the optimal result in both AUROC and average precision score with the value of 0.947 and 0.809, respectively.

Experiment 5	**5.a**	**5.b**	**5.c**
Batch sizes	25	16	16
Epochs	80	80	80
Learning rate	0.001	0.001	0.100
Fraud class MSE	4.978	4.976	5.010
Normal class MSE	0.878	0.867	0.904
AUROC	0.945	0.947	0.948
Avg precision	0.808	0.809	0.804
Acuracy	0.96	0.96	0.96
Sensitivity	0.82	0.81	0.81
Specificity	0.98	0.98	0.98

Table 6: Experiment 5 Performance Results

Figure 13: Reconstruction error of experiment 5b

6. Findings summary and discussion

The performance of the five experiments is summarized as in Figure 14. The models could achieve more than 96% accuracy, 81% sensitivity, and 97% specificity. In most cases, the AUROC values of the proposed model are higher than 0.9. The effects of undersampling and oversample are different for each algorithm.

- For LR algorithms, the model with original features and undersampling method (experiment 3) had the highest results of 0.972 for AUROC and 0.928 for average precision. In contrast, the model with original features and oversampling method had the lowest performance (AUROC: 0.930 and average precision: 0.883). All LR algorithms in the four experiments achieve a high accuracy of more than 97%.
- For KNN algorithms, the models with features selection in experiments 1 and 2 (without applying any sampling method) have the highest performance values of AUROC and the average precision score of 0.933 and 0.865, respectively.
- DT model in experiment 2 with original features has the highest AUROC value of 0.952, while the model applying oversampling method and PCA has the highest average precision value of 0.878 compared to DT models in other experiments.
- RF algorithm has proved to be the best (performing) algorithm for fraud detection with an accuracy of 98% and the highest performance results of 0.976 for AUROC and 0.927 for average precision for the model applying undersampling and feature selection.

Overall, compared to baseline models, the performances of oversampling models are less favorable in terms of AUROC value. The integrating undersampling approach achieves better performance when handling skewed data than the oversampling and adjusting class weights. Except for the DT undersampling model, feature selection using PCA could enhance the AUROC values. RF is the superior performance model across performance metrics, followed by LR. RF with PCA and undersampling achieve the highest AUROC value of 0.976, while LR in full features case and undersampling has the highest average precision score value of 0.928.

Furthermore, autoencoder, unsupervised learning, has consistently achieved good performance with the accuracy of 96% through different batch sizes and learning rates. Besides, its results are better than DT and KNN in terms of AUROC. Finally, DT has the worst results, particularly when applying oversampling method. However, as a trade-off, it has the highest sensitivity, which indicates the ability to detect fraud correctly.

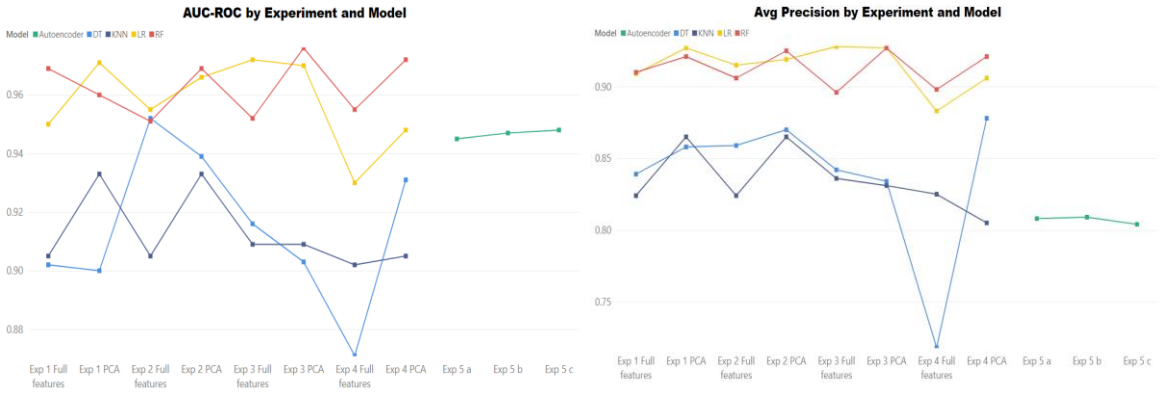


Figure 14: (A) AUROC (B) Average Precision by Experiment and model

VI. CONCLUSION

Digital payment fraud and fraudulent activities have severe impacts on financial businesses and society, particularly with the popularity of mobile payments, digital wallets, payment cards, and contactless cards in the age of Industry 4.0. Significant efforts have been made to tackle such issues. Intelligent approaches such as machine learning algorithms have been successfully utilized and adapted to recognize fraudulent activities automatically for Industry 4.0. This study proposes and evaluates the learning models to detect fraudulent transactions. Therefore, the main contributions are identifying an effective learning model to predict fraud through a comparative study and assessing the proposed models with real credit card activities data. Our study is essential to mitigate the risks of uncertainty and undesirable financial loss suffered by customers and payment providers in the Industry 4.0 era. Secondly, adjustment of hyperparameters automation for supervised learning algorithms is employed to improve fraud classification and reduce times and resources considerably in modeling. There are four supervised learning algorithms applied in the evaluation. In order to estimate the impact of features selection on classification performance, feature engineering and analysis are conducted. Then, the outcomes have been tested with the appraisal of each model addressed. Finally, we also address and propose appropriate solutions for the data skewness matter by considering the effects of oversampling and undersampling techniques. Accuracy is not sensitive to skewed data, so not appropriate as a parameter in this study as it cannot provide a conclusive interpretation. Consequently, we used AUROC and precision as the determining metrics to reach a particular judgment. Various methods were brought together to recognize fraudulent activities, and such comparison of these five models (logistic regression, KNN, decision tree, random forest, and autoencoder)is novel and attractive in literature. Results suggested that the integration undersampling method – NearMiss could improve the models' performance.

Acknowledgment

This work is partly supported by VC Research (VCR 0000158) for Prof Chang.

REFERENCES

[1] B. Machkour and A. Abriane, "Industry 4.0 and its Implications for the Financial Sector," *Procedia Comput. Sci.*, vol. 177, pp. 496–502, Jan. 2020, doi: 10.1016/J.PROCS.2020.10.068.

[2] M. M. Alani and M. Alloghani, "Security Challenges in the Industry 4.0 Era," *Ind. 4.0 Eng. a Sustain. Futur.*, pp. 117–136, 2019, doi: 10.1007/978-3-030-12953-8_8.

[3] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, Nov. 2017, vol. 2017-January, pp. 1–9, doi: 10.1109/ICCNI.2017.8123782.

[4] T. Baabdullah, A. Alzahrani, and D. B. Rawat, "On the Comparative Study of Prediction Accuracy for Credit Card Fraud Detection wWith Imbalanced Classifications," in *Proceedings of the 2020 Spring Simulation Conference, SpringSim 2020*, May 2020, doi: 10.22360/SpringSim.2020.CSE.004.

[5] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection," *Int. J. Inf. Technol.*, 2020, doi: 10.1007/s41870-020-00430-y.

[6] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *J. Inf. Secur. Appl.*, vol. 55, p. 102596, Dec. 2020, doi: 10.1016/j.jisa.2020.102596.

[7] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *ICNSC 2018 - 15th IEEE International Conference on Networking, Sensing and Control*, May 2018, pp. 1–6, doi: 10.1109/ICNSC.2018.8361343.

[8] F. Carcillo, A. Dal Pozzolo, Y. A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with spark," *Inf. Fusion*, vol. 41, pp. 182–194, May 2018, doi: 10.1016/j.inffus.2017.09.005.

[9] S. Mittal and S. Tyagi, "Performance evaluation of machine learning algorithms for credit card fraud detection," in *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019*, Jan. 2019, pp. 320–324, doi: 10.1109/CONFLUENCE.2019.8776925.

[10] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Appl. Energy*, vol. 211, pp. 1123–1135, Feb. 2018, doi: 10.1016/j.apenergy.2017.12.005.

[11] X. Niu, L. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised," Apr. 2019, Accessed: Jul. 03, 2021. [Online]. Available: http://arxiv.org/abs/1904.10604.

[12] F. Fadaei Noghani and M. Moattar, "Ensemble Classification and Extended Feature Selection for Credit Card Fraud Detection," *J. AI Data Min.*, vol. 5, no. 2, pp. 235–243, Jul. 2017, doi: 10.22044/JADM.2016.788.

[13] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.

[14] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, Jun. 2017, doi: 10.1016/J.JII.2017.04.005.

[15] A. Corallo, M. Lazoi, and M. Lezzi, "Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts," *Comput. Ind.*, vol. 114, p. 103165, Jan. 2020, doi: 10.1016/J.COMPIND.2019.103165.

[16] S. C. Dubey, K. S. Mundhe, and A. A. Kadam, "Credit Card Fraud Detection using Artificial Neural Network and BackPropagation," in *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, May 2020, pp. 268–273, doi: 10.1109/ICICCS48265.2020.9120957.

[17] R. Sailusha, V. Gnaneswar, R. Ramesh, and G. Ramakoteswara Rao, "Credit Card Fraud Detection Using Machine Learning," in *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, May 2020, pp. 1264–1270, doi: 10.1109/ICICCS48265.2020.9121114.

[18] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2011*, 2011, Accessed: May 11, 2021. [Online]. Available: https://openaccess.dogus.edu.tr/xmlui/handle/11376/2366.

[19] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011, doi:

10.1016/j.dss.2010.08.008.

[20] A. Mishra and C. Ghorpade, "Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques," in *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2018*, Nov. 2018, doi: 10.1109/SCEECS.2018.8546939.

[21] M. A. Al-Shabi, "Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets," *J. Adv. Math. Comput. Sci.*, vol. 33, no. 5, pp. 1–16, Aug. 2019, doi: 10.9734/jamcs/2019/v33i530192.

[22] A. Alazizi, A. Habrard, F. Jacquenet, L. He-Guelton, and F. Oblé, "Dual Sequential Variational Autoencoders for Fraud Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Apr. 2020, vol. 12080 LNCS, pp. 14–26, doi: 10.1007/978-3-030-44584-3_2.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[24] C. V. Priscilla and D. P. Prabha, "Influence of optimizing xgboost to handle class imbalance in credit card fraud detection," in *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, Aug. 2020, pp. 1309–1315, doi: 10.1109/ICSSIT48917.2020.9214206.

[25] F. Song, Z. Guo, and D. Mei, "Feature selection using principal component analysis," *Proc. - 2010 Int. Conf. Syst. Sci. Eng. Des. Manuf. Informatiz. ICSEM 2010*, vol. 1, pp. 27–30, 2010, doi: 10.1109/ICSEM.2010.14.

Victor Chang received PhD in Computer Science from University of Southampton, UK. He is a Professor in Data Science and Information Systems at Teesside University, UK. He is a Conference Chair of 4 international conferences, Associate Editor/Editor of top journals, Highly Cited 2021 and top 2% Scientist. He won numerous awards and funding, and is an active and influential researcher.

Le Minh Thao Doan is currently a master's student of Applied Data Science at Teesside University. Her expertise is data analytics with solid experience in data analysis, financial and management reporting, and reporting process automation. Her research interest is machine learning and building predictive models to solve business problems and support healthcare, business, and technology decision-making.

Alessandro Di Stefano is currently working as a Lecturer in Computer Science at Teesside University. He received his BSc (2009) and MSc degrees (2012) in Telecommunications Engineering and a Ph.D. degree in Systems Engineering (2015). His main research interests include game theory, network science, and artificial

intelligence. He has published many peer-reviewed papers in high-impact journals and leading international conferences.

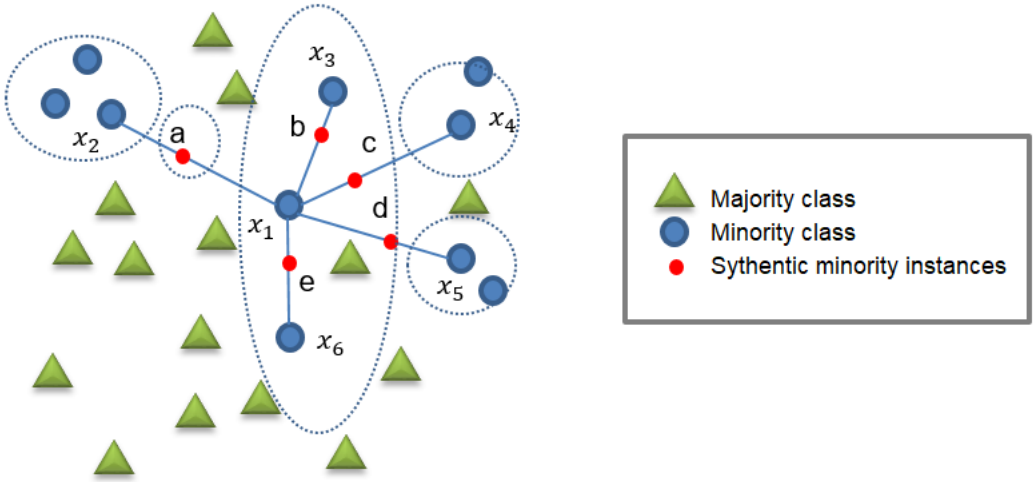
Prof. Zhili Sun is a Chair Professor with 5G&6G Innovation Centre (5G&6GIC), Institute of Communication Systems (ICS), University of Surrey, UK. His research interests include satellite communications and networks, 5G/6G systems, wireless mobile and sensor networks, mobile operating systems, traffic engineering, IP networks and security. He has authored 3 books and published over 240 papers in international journals and conferences.

Giancarlo Fortino is Full Professor of Computer Engineering at the Dept of Informatics, Modeling, Electronics, and Systems of the University of Calabria, Italy. He is IEEE Fellow 2022 and Highly Cited Researcher 2002-2021 in Computer Science. His research interests include wearable computing systems, Internet of Things, and Cyber-security. He is author of 550+ papers in int'l journals, conferences and books.

APPENDICES

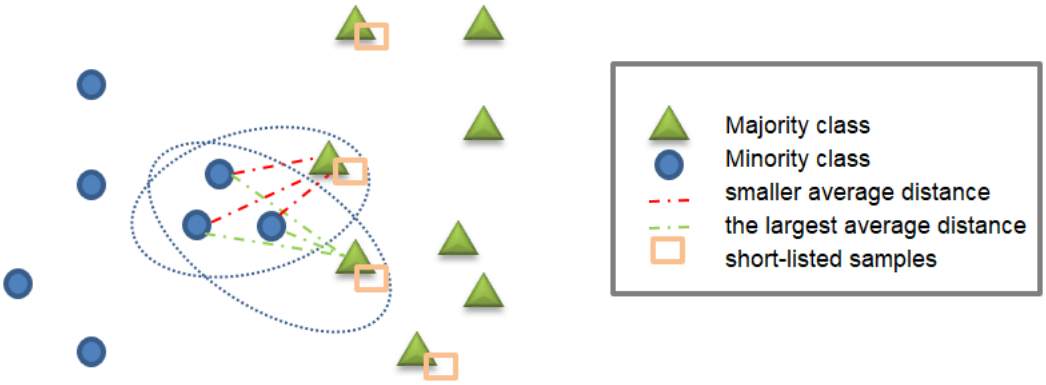
APPENDIX 1: SMOTE Example

The below picture illustrated the application of SMOTE to generate synthetic samples (a, b, c, d, e) for the minority class x_1 by using k =5 nearest neighbor x_2, x_3, x_4, x_5 with the minority class sample x_1.

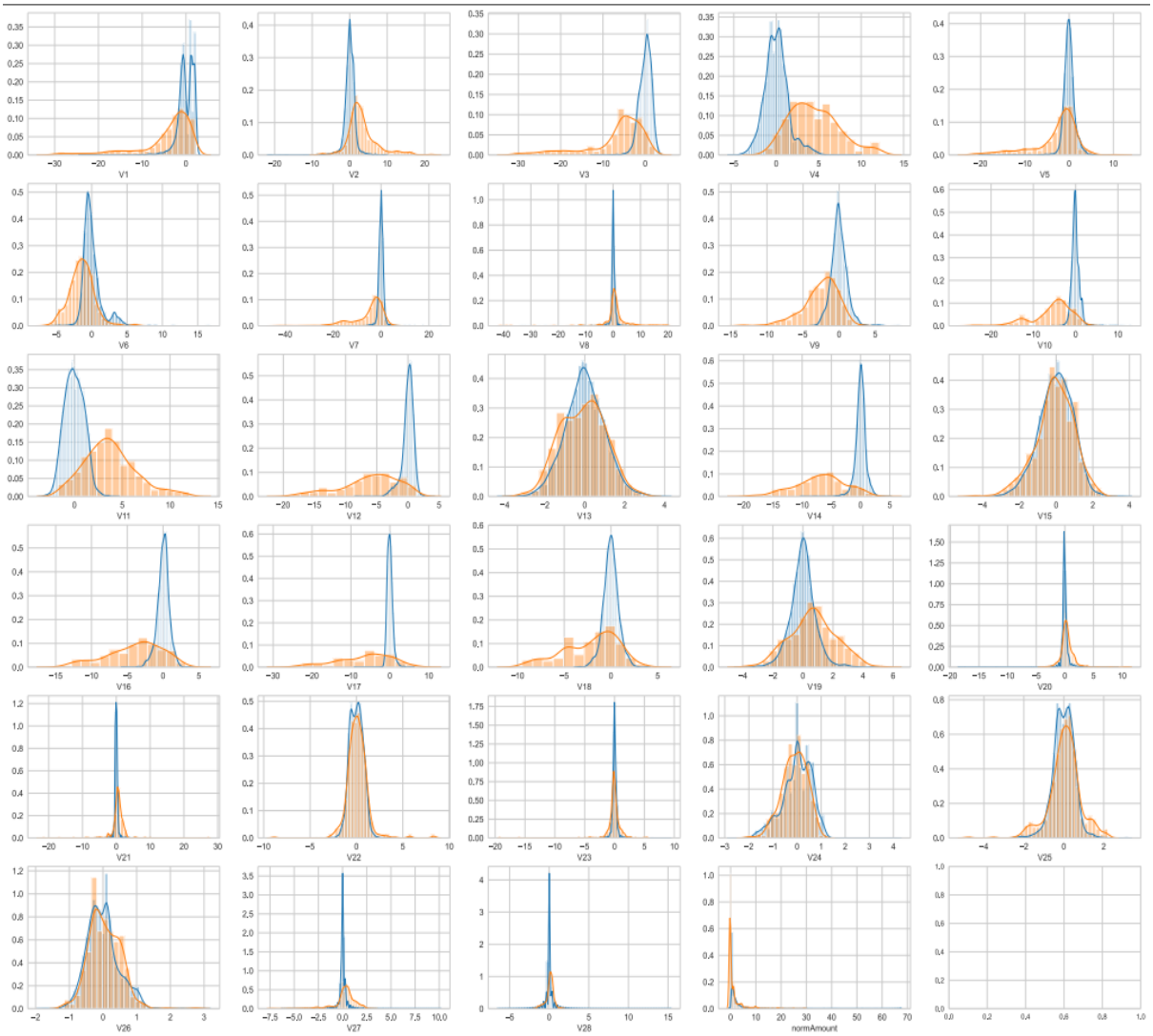


APPENDIX 2: NearMiss Example

We implemented NearMiss (version 3) in our study. This undersampling technique comprised two steps and was presented in the below picture. First, a given nearest number of minority class was chosen for each majority class instance (i.e., corresponding to the highlighted rectangle samples in the below plot with k =3). Next, each short-listed sample's average distance was calculated, and the largest average distance to the k nearest neighbors are selected. In the below example, the majority class with the green dash line was the selected one.



APPENDIX 3: New sample data distribution



APPENDIX 4: New sample data correlation